



TUGAS AKHIR – TI 091324

OPTIMASI NURSE SCHEDULING PROBLEM (STUDI KASUS RSUD DR SOETOMO SURABAYA)

ADITYA PRATAMA HIDAYATULLAH

NRP 2510100111

Dosen Pembimbing

Prof Ir. Budi Santosa, M.S, Ph.D

JURUSAN TEKNIK INDUSTRI

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA 2014



TUGAS AKHIR – TI 091324

OPTIMIZATION OF NURSE SCHEDULING PROBLEM

(CASE STUDY: RSUD DR SOETOMO SURABAYA)

ADITYA PRATAMA HIDAYATULLAH

NRP 2510100111

Supervisor

Prof Ir. Budi Santosa, M.S, Ph.D

DEPARTMENT OF INDUSTRIAL ENGINEERING

FACULTY OF INDUSTRIAL TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA 2014

LEMBAR PENGESAHAN

**OPTIMASI NURSE SCHEDULING PROBLEM (STUDI KASUS RSUD DR
SOETOMO SURABAYA)**

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik
pada

Program Studi S-1 Jurusan Teknik Industri
Fakultas Teknologi Sepuluh Nopember Surabaya

Oleh:

ADITYA PRATAMA HIDAYATULLAH

NRP. 2510100111

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Prof. Ir. Budi Santosa, M.S., Ph.D

.....(Pembimbing)



SURABAYA, JULI 2014

OPTIMASI NURSE SCHEDULING PROBLEM (STUDY KASUS RSUD DR SOETOMO SURABAYA)

Nama Mahasiswa : Aditya Pratama Hidayatullah
NRP : 2510 100 111
Jurusan : Teknik Industri FTI-ITS
Dosen Pembimbing : Prof. Ir. Budi Santosa, M.S., Ph.D

ABSTRAK

Nurse Scheduling Problem (NSP) adalah suatu permasalahan penjadwalan perawat untuk membuat *shift* penjadwalan sejumlah n perawat dengan menentukan satu dari beberapa pola *shift* yang mungkin untuk tiap perawat. Setiap rumah sakit harus dapat melakukan penjadwalan perawat dengan baik agar aktivitas di rumah sakit bisa berjalan dengan baik. Ketidaktepatan dalam melakukan penjadwalan pada perawat, maka perawat tersebut akan bekerja tidak maksimal yang berakibat akan menimbulkan kelelahan secara fisik, tingkat stres yang tinggi, dan rentan melakukan kesalahan dalam melakukan pekerjaan yang akan membahayakan baik dari perawat sendiri maupun pada pasien yang terdapat di rumah sakit. Dalam penyelesaian NSP ini bisa digunakan dengan dua cara, yaitu dengan menggunakan metode eksak dan dengan metaheuristik. Dari kedua metode ini mempunyai kekurangan dan kelebihan sendiri. Pada metode eksak hasil yang didapatkan berada pada titik yang paling optimal (global optimal) tetapi membutuhkan waktu yang sangat lama dalam penyelesaian jika kasus tersebut pada skala yang besar. Sedangkan dengan metaheuristik membutuhkan waktu yang sebentar dalam penyelesaian kasus dengan skala besar pun tetapi hasil yang didapat hanya mendekati solusi optimal saja.

Kata kunci : NSP, minimasi deviasi, *metaheuristik*, metode eksak

OPTIMIZATION FOR NURSE SCHEDULING PROBLEM (CASE STUDY RSUD DR. SOETOMO SURABAYA)

Nama Mahasiswa : Aditya Pratama Hidayatullah
NRP : 2510 100 111
Jurusan : Teknik Industri FTI-ITS
Dosen Pembimbing : Prof. Ir. Budi Santosa, M.S., Ph.D

ABSTRACT

Nurse Scheduling Problem (NSP) is the problem for nurse schedule to make scheduling shift or pattern with n nurse and divide one of many feasible shift pattern for each nurse. Every hospital should make nurse schedule perfectly so the activity in hospital can work well. If the nurse schedule is bad, nurse can not work properly so the impact from it are physical fatigue, stress, and easy to make a mistake for work so it will be dangerous for nurse and patient in hospital. To solve NSP, there are two methods, exact method and metaheuristic. Each method has weakness and benefit. For exact method the result on the global optimum but it need much time to solve if in the large scale. For metaheuristic don't need much time to solve although in the large scale but the result is not optimum, but approach optimum.

Key word : NSP, minimum deviation, metaheuristic method, exact method

KATA PENGANTAR

Segala puji syukur Penulis panjatkan kepada Allah SWT atas segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Optimasi *Nurse Scheduling Problem* (Studi kasus RSUD Dr. Soetomo Surabaya”.

Selama proses pengerjaan Tugas Akhir, penulis telah menerima banyak dukungan, masukan, serta bantuan dari berbagai pihak. Pada kesempatan ini, penulis ingin mengucapkan terimakasih kepada:

1. Kedua orang tua penulis, yaitu Lili Aliyudin SE dan Ernawati yang selalu mendukung penuh dan semangat penulis dalam penyelesaian tugas akhir ini.
2. Kedua adik saya yaitu Anindya dan Akmal yang selalu sebagai bahan usil ketika lagi suntuk saat menyelesaikan tugas akhir.
3. Prof. Ir. Budi Santora, M.S., Ph.D selaku Ketua Jurusan Teknik Industri, Kepala Laboratorium Komputasi dan Optimasi Industri, serta sebagai bapak dosen pembimbing penulis, terima kasih atas semua pembelajaran dan arahan kepada penulis dalam penyelesaian tugas akhir ini.
4. Ibu Muzhidah selaku kepala keperawatan RSUD Dr. Soetomo Surabaya yang memberikan kemudahan kepada penulis dalam pengambilan data yang dibutuhkan dalam tugas akhir ini.
5. Kalapie Astragina, yang selalu memberikan perhatian khusus kepada penulis agar cepat menyelesaikan tugas akhir ini.
6. Anak-anak bimbingan Pak Budi Santosa, Gusti, Risal, Abdan, Sophia, dan Irfan yang selalu membantu penulis dan memperlancar pengerjaan tugas akhir ini.
7. KWU 11/12 dan KWU 12/13 terimakasih atas segala pengalaman, pembelajaran, dan rasa kekeluargaan yang dalam di organisasi kepada penulis.
8. Keluarga besar der Administrator Laboratorium Komputasi dan Optimasi Industri dari angkatan 2010 : Azul, Gusti, Andrew, Hasyim, Vega, Layli,

Sabrina, Dewi, Puhenk, Hajar angkatan 2011: Aan, Resa, Chrisman, Nindya, Mike, Friska, Ovita, Lola, Agni, dan mas mbak S2 : Mas Fiqi, Mas Angga dan Mbak Eki atas rasa kekeluargaan dan kebersamaan selama ini.

9. Semua Asisten PSI yang selalu saling mendukung dalam pengerjaan dan penyelesaian tugas akhir ini.
10. Geng Sepeda-UniseX(Unifier Sepeda 10) yang menjadi tempat untuk refreshing penulis bersepeda di tempat-tempat seru ketika suntuk dalam penyelesaian tugas akhir ini.
11. Teman-teman aksel libels yang selalu sebagai alternatif lain buat seru-seruan dan nongkrong bersama penulis ketika lagi bermalas-malasan saat penyelesaian tugas akhir ini.
12. Teman-temen provokasi 2010. Terima kasih atas kebersamaan, pengalaman, dan kekeluargaan selama 4 tahun ini. Cepat lulus semua yaaa.

Serta berbagai pihak yang tidak dapat penulis sebut satu persatu, terima kasih atas semua dukungan, nasihat, semangat yang selalu diberikan kepada penulis, semoga Allah SWT membalas semua kebaikan tersebut. Amiin.

Penulis menyadari bahwa pengerjaan Tugas Akhir ini masih jauh dari sempurna. Kritik dan saran sangat penulis butuhkan untuk perbaikan ke depannya. Penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi para pembacanya. Sekian yang dapat penulis sampaikan, akhir kata penulis mengucapkan terima kasih.

Surabaya, Juli 2014

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	6
1.3 Ruang Lingkup Penelitian	6
1.3.1 Batasan	6
1.3.2 Asumsi.....	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
1.6 Sistematika Penulisan	7
BAB 2.....	9
TINJAUAN PUSTAKA.....	9
2.1 Konsep Penjadwalan.....	9
2.2 Konsep Keperawatan.....	9
2.3 Nurse Scheduling Problem	10
2.3.1 <i>Hard constraint</i>	11
2.3.2 <i>Soft constraint</i>	11
2.4 Model NSP.....	12
2.4.1 Parameter.....	12
2.4.2 Variabel Keputusan	12
2.4.3 <i>Hard Constraint</i>	13
2.4.4 <i>Soft Constraint</i>	14
2.4.5 Goal	15

2.4.6	Fungsi Tujuan.....	16
2.5	Simulated Annealing.....	16
2.6	Literature Review.....	19
BAB 3	21
METODOLOGI PENELITIAN	21
3.1	Pengumpulan Data.....	22
3.2	Penyusunan Model Matematis <i>Nurse Scheduling Problem</i>	22
3.2.1	Parameter.....	22
3.2.2	Variabel Keputusan.....	23
3.2.3	<i>Hard Constraint</i>	23
3.2.4	<i>Soft Constraint</i>	24
3.2.5	<i>Goal</i>	24
3.2.6	Bobot Kepentingan Fungsi Tujuan.....	25
3.2.7	Fungsi Tujuan.....	26
3.3	Penyusunan Algoritma <i>Simulated Annealing</i>	26
3.3.1	Penentuan Parameter.....	28
3.3.2	Bangkitkan solusi awal dan nilai fungsi tujuan solusi awal.....	28
3.3.3	<i>Update</i> iterasi dan siklus.....	29
3.3.4	Pembangkitan solusi baru dan nilai fungsi tujuan solusi baru.....	29
3.3.5	Membandingkan nilai fungsi tujuan solusi baru dengan nilai fungsi tujuan solusi sebelumnya.....	29
3.3.6	<i>Stopping Criteria</i>	30
3.4	Validasi Model dan Algoritma.....	30
3.5	Pembuatan Kode Program.....	31
3.6	Verifikasi.....	31
3.7	Eksperimen.....	31
3.8	Analisis.....	32
3.9	Kesimpulan dan Saran.....	32
BABIV	33

PENGUJIAN ALGORITMA.....	33
4.1 Verifikasi dan Validasi Model Matematis	33
4.2 Langkah-Langkah dalam <i>Simulated Annealing</i>	37
4.3 Verifikasi dan Validasi Algoritma	47
BAB V	49
EKSPERIMEN DAN ANALISIS	49
5.1 Deskripsi Data Uji	49
5.2 Eksperimen	50
5.2.1 Eksperimen dengan Algoritma <i>Simulated Annealing</i>	51
5.2.1.1 Uji Parameter Faktor Pereduksi Temperatur (Cr)	51
5.2.1.2 Uji Parameter Siklus Penurunan Temperatur(N).....	53
5.2.1.3 Uji Parameter Temperatur Awal (To).....	54
5.2.1.4 Eksperimen Algoritma <i>Simulated Annealing</i>	56
5.2.2 Eksperimen dengan Menggunakan Metode Eksak.....	57
5.3 Perhitungan Kondisi Eksisting	59
5.4 Perbandingan Hasil Pengujian	60
5.4.1 Perbandingan Deviasi Pelanggaran Kondisi Eksisting dengan Algoritma <i>Simulated Annealing</i>	60
5.4.2 Perbandingan Deviasi Pelanggaran dengan Metode Eksak dan Kondisi Eksisting	64
5.4.3 Perbandingan Deviasi Pelanggaran dengan Algoritma <i>Simulated</i> <i>Annealing</i> dan Metode Eksak.....	65
5.5 Analisis Hasil Eksperimen.....	69
5.5.1 Analisis Hasil Eksperimen dengan Algoritma <i>Simulated Annealing</i>	69
5.5.1.1 Analisis Uji Parameter pada Algoritma <i>Simulated Annealing</i>	69
5.5.1.2 Analisis Hasil Eksperimen dengan Algoritma <i>Simulated Annealing</i>	71
5.5.2 Analisis Eksperimen dengan Menggunakan Metode Eksak	72
5.5.3 Analisis Perbandingan Deviasi Pelanggaran dengan <i>Simulated</i> <i>Annealing</i> dan Kondisi Eksisting	72

5.5.4	Analisis Perbandingan Deviasi Pelanggaran dengan Metode Eksak dan Kondisi Eksisting	73
5.5.5	Analisis Perbandingan Deviasi Pelanggaran dengan Algoritma <i>Simulated Annealing</i> dan Menggunakan Metode Eksak	74
BAB VI		76
KESIMPULAN DAN SARAN		76
5.6	Kesimpulan	76
5.7	Saran	77
DAFTAR PUSTAKA		78
LAMPIRAN A		80
LAMPIRAN B		92
BIODATA PENULIS		106

DAFTAR TABEL

Tabel 2.1 <i>Literature Review</i>	19
Tabel 3. 1 Pembangkitan solusi awal.....	28
Tabel 4. 1 Data contoh kasus kecil.....	33
Tabel 4. 2 Pola Penjadwalan	34
Tabel 4. 3 Data Contoh Kasus Kecil	38
Tabel 4. 4 Solusi Awal <i>Shift</i> Malam.....	39
Tabel 4. 5 Solusi Awal Contoh Kasus Kecil	41
Tabel 4. 6 Solusi Awal dan Nilai Fungsi Tujuan dari Solusi Awal	41
Tabel 4. 7 Solusi Baru	43
Tabel 4. 8 Solusi Baru dan Nilai Fungsi Tujuan Solusi Baru	43
Tabel 5. 1 Rincian Kebutuhan Perawat Tiap <i>Shift</i> Tiap Harinya.....	49
Tabel 5. 2 Rincian <i>Workload</i> Tiap Perawat Tiap <i>Shift</i>	50
Tabel 5. 3 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.5.....	51
Tabel 5. 4 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.7.....	51
Tabel 5. 5 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr)=0.9.....	52
Tabel 5. 6 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.95.....	52
Tabel 5. 7 Deviasi Pelanggaran dengan Jumlah Siklus Penurunan Temperatur (N) = 10.....	53
Tabel 5. 8 Deviasi Pelanggaran dengan Jumlah Siklus Penurunan Temperatur (N) = 15.....	53
Tabel 5. 9 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 2000.....	54
Tabel 5. 10 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 5000.....	55
Tabel 5. 11 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 15000.....	55
Tabel 5. 12 Rincian Pelanggaran Uji Parameter dengan <i>Stopping Criteria</i> Temperatur sebesar 10^{-6}	56
Tabel 5. 13 Rincian Pelanggaran Uji Parameter dengan <i>Stopping Criteria</i> Jumlah Iterasi sebesar 50.000.....	57
Tabel 5. 14 Rincian Deviasi Pelanggaran pada Metode Eksak (1).....	58
Tabel 5. 15 Rincian Deviasi Pelanggaran Kondisi Eksisting (1).....	59
Tabel 5. 16 Rincian Deviasi Pelanggaran Kondisi Eksisting (2).....	59
Tabel 5. 17 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting pada <i>Workload Shift Pagi</i>	60
Tabel 5. 18 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting <i>Workload Shift sore</i>	61
Tabel 5. 19 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting <i>Workload Shift Malam</i>	61

Tabel 5. 20 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Workload Shift Libur.....	62
Tabel 5. 21 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift Sore Diikuti Shift Pagi.....	62
Tabel 5. 22 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift Pagi Diikuti Shift Malam.....	63
Tabel 5. 23 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift "Off-On-Off".....	63
Tabel 5. 24 Perbandingan Total Deviasi Pelanggaran yang Terjadi.....	64
Tabel 5. 25 Perbandingan Metode Eksak dan Kondisi Eksisting.....	64
Tabel 5. 26 Perbandingan Metode Eksak dan Kondisi Eksisting (cont).....	65
Tabel 5. 27 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift Pagi.....	65
Tabel 5. 28 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift sore.....	66
Tabel 5. 29 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift Malam.....	66
Tabel 5. 30 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift Libur.....	67
Tabel 5. 31 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift Sore Diikuti Shift Pagi.....	67
Tabel 5. 32 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift Pagi Diikuti Shift Malam.....	68
Tabel 5. 33 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift "Off-On-Off".....	68
Tabel 5. 34 Perbandingan SA dan Metode Eksak pada Waktu Komputasi.....	69

DAFTAR GAMBAR

Gambar 2. 1 <i>flowchart simulated annealing</i>	18
Gambar 3. 1 <i>Flowchart</i> penelitian.....	21
Gambar 3. 2 Algoritma <i>Simulated Annealing</i>	27
Gambar 4. 1 Hasil Contoh Kasus Kecil Menggunakan Lingo.....	35
Gambar 4. 2 Hasil <i>Output</i> Algoritma <i>Simulated Annealing</i> (1).....	45
Gambar 4. 3 Hasil <i>Output</i> Algoritma <i>Simulated Annealing</i> (2).....	46
Gambar 4. 4 Grafik <i>Output</i> Penyelesaian dengan Algoritma <i>Simulated Annealing</i>	46
Gambar 5. 1 Status Eksperimen dengan <i>Software</i> LINGO.....	58

BABI

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang mendasari dilakukannya penelitian serta identifikasi masalah penelitian yang meliputi latar belakang, perumusan masalah, ruang lingkup permasalahan yang terdiri dari batasan dan asumsi, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

1.1 Latar Belakang

Penjadwalan adalah proses pengorganisasian, pemilihan, dan penentuan waktu penggunaan sumber-sumber untuk mengerjakan semua aktivitas yang diperlukan yang memenuhi kendala aktivitas dan sumber daya (Morton and Pentico, 1993). Sumber daya yang dijadwalkan bisa berupa perlengkapan, peralatan, fasilitas, dan tenaga kerja. Penjadwalan tenaga kerja merupakan salah satu hal penting yang harus diperhatikan di dalam dunia industri baik industri di bidang jasa maupun industri di bidang manufaktur. Hal itu dikarenakan penjadwalan tenaga kerja akan mempengaruhi kinerja dan produktifitas dari industri tersebut baik secara langsung maupun secara tidak langsung. Sebagai contoh jika penjadwalan tenaga kerja yang buruk maka jika dilihat dari segi biaya akan bertambah akibat tidak maksimalnya penjadwalan tenaga kerja tersebut dan produktifitas dari industri tersebut tidak maksimal karena buruknya penjadwalan tenaga kerja.

Adapun salah satu penjadwalan tenaga kerja yang sangat berpengaruh pada aktivitas pekerjaan adalah rumah sakit. Rumah sakit merupakan sebagai salah satu sarana kesehatan yang memberikan pelayanan kesehatan kepada masyarakat memiliki peran yang sangat strategis dalam mempercepat peningkatan derajat kesehatan masyarakat. Adapun sumber daya manusia yang terdapat di rumah sakit yaitu dokter, perawat, bidan, dan bagian farmasi. Sumber daya manusia tersebut merupakan bagian-bagian yang selalu bersinggungan dengan pasien sehingga baik buruknya kinerja dari rumah sakit akan bergantung dengan sumber daya manusia tersebut. Dari sumber daya manusia yang terdapat di rumah

sakit, perawat merupakan salah satu sumber daya yang selalu bersinggungan dengan pasien. Selama 24 jam setiap hari dan 7 hari setiap minggu, perawat selalu *stand by* di rumah sakit untuk memberikan pelayanan kepada pasiennya.

Performansi rumah sakit bisa dilihat pasien dari bagaimana seorang perawat bisa selalu ada setiap pasien tersebut membutuhkannya dan bagaimana seorang perawat bisa memberlakukan pasien dengan baik. Karena hal tersebut maka diperlukan penjadwalan perawat di rumah sakit dengan sangat baik agar perawat bisa bekerja dengan maksimal. Hal ini dikarenakan jika penjadwalan perawat ini buruk misalnya seorang perawat telah berjaga seharian penuh maka kondisi perawat tersebut akan menurun akibat terjadinya kelelahan secara fisik, stres yang tidak semestinya terjadi, dan rentan melakukan kesalahan pada pekerjaan (M'Hallah & Alkhabbaz, 2013). Karena penjadwalan perawat yang buruk itu maka perawat tersebut akan memberikan pelayanan yang buruk kepada pasiennya sehingga akan menurunkan performansi pelayanannya dan imbasnya akan memperburuk citra dari rumah sakit tersebut. Karena alasan tersebut, peran dari perawat merupakan hal yang vital bagi kinerja dari rumah sakit sehingga untuk menghindari hal-hal tersebut maka harus dilakukan penjadwalan perawat dengan baik dengan menggunakan suatu metode dalam penjadwalan tersebut yang dikenal dengan *nurse scheduling problem*(NSP).

RSUD Dr. Soetomo Surabaya merupakan rumah sakit terbesar di Jawa Timur dan merupakan rumah sakit rujukan pasien dari rumah sakit di kota lain sehingga terdapat banyak sekali pasien yang ditangani oleh rumah sakit ini, baik dari wilayah Surabaya, juga pasien rujukan dari rumah sakit di luar Kota Surabaya. Tercatat pada akhir tahun 2013 terdapat hampir 200.000 pasien yang ditangani oleh RSUD Dr. Soetomo Surabaya. Dikarenakan banyaknya pasien yang harus ditangani dan cepatnya pasien keluar dan masuk rumah sakit sehingga pelayanan dari rumah sakit harus benar-benar diperhatikan. Salah satu cara agar mendapatkan pelayanan yang terbaik yaitu dengan melakukan penjadwalan sebaik mungkin pada sumber daya manusia yang ada. Salah satu penjadwalan sumber daya manusia di rumah sakit adalah penjadwalan perawat. Di RSUD Dr. Soetomo dalam melakukan penjadwalan perawat dilakukan oleh tiap kepala ruangan dengan metode konvensional. Terdapat 3 *shift* kerja di tiap harinya, yaitu *shift*

pagi, sore, dan malam. Dikarenakan masih menggunakan metode yang konvensional, masih banyak pelanggaran yang tidak sesuai dengan aturan di RSUD Dr. Soetomo Surabaya pada hasil penjadwalan tersebut. Sebagai contoh masih terdapat perawat yang tidak mendapatkan salah satu *shift* yang ada, padahal menurut aturan dari kepegawaian rumah sakit tiap perawat harus mempunyai *workload* yang sama dalam mendapatkan *shift* kerja. Selain itu penjadwalan di RSUD Dr. Soetomo yang dilakukan tiap kepala ruangan masih sangat fleksibel sehingga kapanpun walaupun perawat sudah terjadwal mendapatkan *shift* jaga tertentu masih bisa tidak masuk dalam *shift* tersebut melalui perijinan oleh kepala ruangan dan kepala ruangan akan mencari pengganti perawat lain untuk menggantikan *shift* tersebut. Karena penjadwalan yang fleksibel membuat penjadwalan perawat tersebut menjadi sangat tidak teratur. Hal itu juga menimbulkan kekhawatiran oleh kepala keperawatan RSUD Dr. Soetomo dikarenakan pada suatu waktu akan menyulitkan pencarian perawat pengganti ketika semua perawat tidak dapat menggantikan jadwal tersebut dan ruangan tersebut akan kosong karena tidak terdapat perawat yang bisa dan akan berbahaya bagi pasien-pasien yang membutuhkan pertolongan pertama dari perawat. Oleh karena itu harus dilakukan perbaikan dalam penjadwalan perawat yang terdapat pada RSUD Dr. Soetomo Surabaya ini agar tiap perawat akan mendapatkan *workload* yang sama sehingga tidak akan menimbulkan kelelahan fisik yang didera oleh perawat sehingga performansi pelayanan dari RSUD Dr. Soetomo Surabaya akan tetap terjaga.

Menurut Ko et al.(2013) NSP membuat suatu penjadwalan secara periodik, apakah itu mingguan atau bulanan untuk setiap n perawat dengan menugaskan atau menempatkan salah satu dari tiap *shift* yang mungkin untuk setiap perawat dengan mempertimbangkan *constraint* yang ada baik *hard* atau *soft constraints*. Solusi dari penjadwalan harus memenuhi *hard constraint* dimana *constraint* tersebut harus dipenuhi dan tidak boleh dilanggar, contohnya adalah jumlah perawat yang harus ada tiap *shift* di tiap harinya. Lalu *soft constraint* dimana *constraint* bisa atau boleh dilanggar tetapi harus seminimal mungkin agar harapannya agar mendapatkan hasil penjadwalan yang baik, contohnya adalah jumlah hari libur, *shift* malam, *shift* sore, *shift* pagi di tiap perawat.

Dengan kemajuan dari teknologi, NSP yang sebelumnya menggunakan metode yang konvensional maka saat ini dapat dilakukan penjadwalan dengan otomatisasi sehingga akan memberikan hasil yang cepat dan optimal. Terdapat beberapa pendekatan dalam menemukan solusi penjadwalan, yaitu pendekatan *artificial intelligence*, *constraint programming*, *mathematical programming*, dan *metaheuristic method* (A.T. Ernst, 2004). Dikarenakan NSP merupakan permasalahan yang kompleks sehingga dikategorikan sebagai NP-Hard. Kompleksitas ini dikarenakan banyaknya jumlah variabel atau parameter-parameter didalamnya seperti jumlah perawat, jumlah *shift*, batasan-batasan yang mungkin, dan parameter-parameter yang terdapat pada fungsi tujuannya. Dengan kompleksitas tersebut jika diselesaikan dengan metode eksak maka kelemahannya waktu yang dibutuhkan akan menjadi sangat lama tetapi hasil yang didapatkan merupakan solusi global optimal. Sementara jika NSP diselesaikan dengan menggunakan metaheuristik hanya membutuhkan waktu yang sebentar dalam penyelesaiannya tetapi solusi yang didapatkan hanya mendekati optimal saja. Sehingga Dengan menggunakan kedua metode optimasi ini dalam penyelesaian NSP, agar dapat mengetahui bagaimana performansi yang didapatkan dari kedua metode optimasi dalam penyelesaian NSP ini.

Metode eksak merupakan salah satu metode penyelesaian dalam metode optimasi dimana dalam penyelesaian suatu masalah akan selalu menemukan solusi yang paling optimal. Tetapi kelemahan dari metode eksak ini digunakan dalam penyelesaian dengan masalah yang besar akan membutuhkan waktu yang sangat lama dalam penyelesaiannya, tetapi hasil yang didapat merupakan solusi yang global optimal.

Metaheuristic adalah metoda untuk mencari solusi yang memadukan interaksi antara prosedur pencarian local dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik *local optimum* dan melakukan pencarian di ruang solusi untuk menemukan solusi global (Santosa, 2011). Dalam penyelesaian permasalahan menggunakan *metaheuristic* tidak akan menjamin hasil yang didapat adalah solusi yang optimal tetapi sudah mendekat solusi yang optimal. Salah satu metode yang digunakan dalam metaheuristik adalah *simulated annealing*. *Simulated annealing* merupakan salah satu metode

dalam metode *metaheuristic* yang bisa digunakan untuk NSP pada rumah sakit. *Simulated annealing* ini termasuk algoritma yang meniru perilaku fisik dari proses pendinginan baja. Ketika baja dipanaskan hingga suhu mendidih maka atom-atom pada baja akan bergerak bebas dan ketika suhu mulai turun maka pergerakan atom akan terbatas sehingga akan menimbulkan susunan atom yang baik. *Simulated annealing* meniru proses pendinginan perlahan besi atau baja yang mendidih. Proses pendinginan dari besi atau baja tersebut dikontrol dengan menggunakan konsep distribusi probabilitas *Boltzmann* yang menyatakan bahwa energy dari suatu sistem dalam keseimbangan panas pada suhu T terdistribusi secara probabilistik. Jika pada permasalahan menggunakan metode *simulated annealing* ini memiliki suhu yang tinggi yaitu suhu awal maka peluang untuk mendapatkan perbedaan energi antara iterasi yang baru dengan yang sebelumnya akan besar. Tetapi dengan menurunnya suhu pada iterasi-iterasi berikutnya maka hasil yang didapatkan untuk iterasi-iterasi selanjutnya akan semakin baik.

Jadi untuk mengganti penyelesaian dalam penyelesaian NSP yang masih menggunakan cara yang konvensional, maka dengan kemajuan teknologi maka akan dilakukan optimasi NSP secara terotomasi, yaitu dengan menggunakan metode eksak dan metaheuristik dimana dengan menggunakan kedua metode ini dapat melihat bagaimana performansi dari kedua metode tersebut.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya, maka permasalahan yang akan dibahas dalam penelitian ini adalah optimasi *nurse scheduling problem* dengan tujuan minimasi deviasi terhadap aturan penjadwalan dari rumah sakit

1.3 Ruang Lingkup Penelitian

Dalam ruang lingkup penelitian ini akan dijelaskan mengenai batasan dan asumsi yang digunakan dalam penelitian ini.

1.3.1 Batasan

Berikut merupakan batasan yang digunakan di dalam penelitian ini :

1. Menggunakan data uji dari data penjadwalan perawat pada RSUD Dr. Soetomo Surabaya
2. Data berasal dari data perawat IRNA Medik
3. Data perawat yang digunakan merupakan perawat tetap di RSUD Dr. Soetomo Surabaya

1.3.2 Asumsi

1. Tidak ada waktu transisi dari *shift* awal dengan *shift* selanjutnya
2. Perawat di *shift* tersebut melakukan pekerjaannya secara *continuous*
3. Kebutuhan perawat dengan jumlah pasien yang ada adalah sama
4. Tidak ada perawat yang mengambil cuti saat dilakukan penelitian ini

1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai dalam penelitian ini adalah

1. Penyelesaian *nurse scheduling problem* dengan metode optimasi yaitu metode eksak dan metaheuristic
2. Mendapatkan hasil NSP dengan tujuan minimasi deviasi terhadap aturan penjadwalan dari rumah sakit

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah untuk mendapatkan penjadwalan perawat yang lebih optimal dengan menggunakan metode optimasi dan dapat digunakan sebagai dasar pengembangan pada sistem pendukung rumah sakit tersebut dalam melakukan proses penjadwalan perawat

1.6 Sistematika Penulisan

1. BAB I PENDAHULUAN

Pada bab ini berisi tentang hal-hal yang mendasari penelitian ini beserta identifikasi masalah penelitian yang berisi latar belakang penelitian, perumusan masalah, ruang lingkup penelitian, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang uraian teori dari permasalahan dan metode yang akan digunakan, penemuan, dan bahan hasil penelitian

lainnya yang diperoleh dari referensi yang akan digunakan sebagai landasan dalam kegiatan penelitian dalam tugas akhir.

3. BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang metodologi penelitian yang akan dilakukan dalam pengembangan algoritma metode *simulated annealing* untuk menjadwalkan perawat di rumah sakit Dr. Soetomo Surabaya.

4. BAB IV PENGUJIAN ALGORITMA

Pada bab ini berisi tentang validasi dari modifikasi model algoritma metode *simulated annealing* dengan metode *simulated annealing* secara umum pada contoh kasus yang sederhana dan berikutnya dilakukan perbandingan dengan perhitungan eksak. Setelah itu dilakukan verifikasi kode program dari keduanya.

5. BAB V EKSPERIMEN DAN ANALISIS

Pada bab ini berisi tentang eksperimen dengan menggunakan beberapa data uji dan analisis dari hasil eksperimen yang telah dilakukan.

6. BAB VI KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan dari hasil penelitian beserta saran-saran yang terkait dengan penelitian selanjutnya.

BAB 2

TINJAUAN PUSTAKA

1.1 Konsep Penjadwalan

Penjadwalan adalah pengalokasian waktu yang tersedia untuk melaksanakan masing-masing pekerjaan dalam rangka menyelesaikan suatu kegiatan hingga tercapai hasil yang optimal dengan mempertimbangkan keterbatasan-keterbatasan yang ada (Husen, 2008).

Penjadwalan tenaga kerja dikategorikan sebagai hal yang cukup penting untuk diperhatikan karena memiliki karakteristik yang spesifik dan kompleks, yaitu

1. Kebutuhan karyawan yang berfluktuasi
2. Tenaga kerja yang tidak bisa disimpan
3. Faktor kenyamanan pelanggan

Sehingga dengan adanya karakteristik tersebut, penjadwalan tenaga kerja menjadi hal yang sangat penting bagi perusahaan agar dapat mengoptimalkan tenaga kerja yang dimilikinya.

Terdapat beberapa manfaat dalam melakukan penjadwalan, antara lain

1. Memberikan pedoman terhadap pekerjaan mengenai batas-batas waktu untuk mulai dan berakhir dari pekerjaan tersebut
2. Sebagai alat untuk mengkoordinir secara sistematis dan realistis dalam penentuan alokasi prioritas terhadap sumber daya dan waktu
3. Menghindari pemakaian sumber daya yang berlebihan
4. Memberikan kepastian waktu pelaksanaan pekerjaan
5. Sebagai sarana untuk melakukan penilaian terhadap kemajuan dari pekerjaan

1.2 Konsep Keperawatan

Keperawatan adalah suatu bentuk layanan kesehatan profesional yang merupakan bagian integral dari layanan kesehatan yang berlandaskan ilmu dan kiat keperawatan yang berbentuk layanan bio, psiko, social, dan spiritual yang komprehensif yang ditujukan bagi individu, keluarga, dan masyarakat baik dalam

keadaan sehat maupun sakit serta mencakup seluruh proses kehidupan.(Asmadi, 2008).

Perawat merupakan tenaga kerja yang dominan dan harus ada di setiap rumah sakit untuk memberikan pelayanan kesehatan tidak hanya kepada pasien, tetapi juga kepada keluarga pasien yang menunggu. Terdapat 3 fungsi perawat secara garis besar, yaitu

1. Fungsi independen dimana perawat tidak memerlukan perintah dari dokter sehingga tindakan perawat bersifat mandiri yang berdasarkan ilmu tindakan keperawatan.
2. Fungsi interindependen dimana tindakan perawat berdasarkan pada kerjasama dengan tim keperawatan atau petugas medis lainnya
3. Fungsi dependen dimana perawat bertindak untuk membantu dokter dalam melakukan pelayanan medis

1.3 Nurse Scheduling Problem

Nurse Scheduling Problem (NSP) adalah suatu permasalahan penjadwalan perawat untuk membuat *shift* penjadwalan sejumlah n perawat dengan menentukan satu dari beberapa pola *shift* yang mungkin untuk tiap perawat (Ko, 2013). Dalam melakukan penjadwalan perawat terdapat istilah *nurse roster*. Menurut Kundu et al (2008) *nurse rostering* adalah ketersediaan waktu antara *shift* penjadwalan dengan waktu istirahat per hari dari perawat tersebut. Tujuan dari *nurse rostering* untuk mendapatkan *shift* penjadwalan yang terbaik tanpa mengurangi kesejahteraan dari perawat tersebut. Jadi penjadwalan yang dibuat juga harus cukup adil bagi perawat tanpa mengganggu kesehatan, keluarga, dan kehidupan sosialnya sehingga hal tersebut menjadi pertimbangan dalam melakukan penjadwalan perawat.

Dalam penyelesaian NSP dilihat dari bagaimana cara dari pihak manajemen rumah sakit ini untuk membuat suatu penjadwalan untuk perawat yang terdapat pada rumah sakit tersebut. Hasil dari penjadwalan ini akan berpengaruh dari kinerja dan loyalitas yang diberikan oleh perawat tersebut. Sebagai contoh jika pada penjadwalan dalam satu periode tertentu seorang perawat mendapatkan jadwal jaga terus-menerus pada *shift* malam, maka akan

memberikan dampak negatif bagi perawat tersebut yang mungkin akan membuat perawat itu tidak bisa fokus dalam bekerja atau berjaga di rumah sakit karena selalu berjaga pada malam hari dan menyebabkan kondisi perawat yang akan melemah akan hal tersebut. Selain itu jika penjadwalan perawat tidak tepat sebagai contoh ada seorang perawat pada hari pertama dan kedua berjaga pada *shift* malam sedangkan pada hari ketiga berjaga *shift* pagi maka akan berdampak pada kinerja dan performansi dari perawat yang tidak optimal karena harus berjaga berturut-turut. Sehingga dalam penjadwalan harus dilakukan dengan baik dan tepat agar tidak berdampak pada performansi dan kinerja pada perawat tersebut.

Pada NSP ini, ada banyak *constraints* yang harus diperhatikan dalam penyelesaian permasalahan tersebut. *Constraints* tersebut dapat diklasifikasikan sebagai *hard* dan *soft constraints*.

1.3.1 Hard constraint

Hard constraint merupakan batasan yang harus dipenuhi dalam melakukan penjadwalan dan tidak diperbolehkan adanya pelanggaran pada batasan tersebut. Jika pada penjadwalan tidak dipenuhi batasan tersebut maka tidak mendapatkan solusi yang *feasible*. Contoh dari batasan ini adalah tiap perawat hanya dapat mengisi 1 *shift* perhari.

1.3.2 Soft constraint

Soft constraint merupakan batasan yang bisa dilanggar, tapi seminimal mungkin agar mendapatkan hasil yang optimal. Sebenarnya *soft constraint* ini harus dipenuhi, tapi jika terdapat pelanggaran maka solusi yang keluar tetap *feasible*. Pada *soft constraint* ini terdapat *threshold* yaitu batas antara pelanggaran yang dapat diterima dengan tidak diterima. Contoh dari batasan ini adalah waktu optimal dari perawat bekerja sebesar 40 jam tiap minggu, tetapi batas yang masih diterima adalah antara 25-60 jam bekerja tiap minggu.

1.4 Model NSP

Dalam penelitian yang dilakukan oleh Jenal Ruzzakiah et al (2011) untuk mengembangkan model dari NSP berdasarkan tujuan atau peraturan dari rumah sakit tersebut dan kecenderungan dan pertimbangan dari perawat tersebut.

Pengembangan model dari NSP berdasarkan tujuan dari rumah sakit berisi *hard constraint* dimana semua batasan yang ada harus di tercapai. Sedangkan jika berdasarkan kecenderungan atau pertimbangan dari perawat tersebut berisi *soft constraint* dimana batasan yang ada boleh dilanggar tetapi seminimal mungkin.

Penyelesaian NSP ini akan dipisah menjadi 2 grup dimana grup pertama berisi dari *hard constraint* dimana dalam penyelesaian NSP tersebut harus sesuai dengan *hard constraint* yang ada sehingga akan mendapatkan hasil yang *feasible*. Kemudian setelah mendapatkan hasil tersebut maka ditinjau kembali dari *soft constraint* yang ada untuk mendapatkan pelanggaran yang seminimal mungkin dari hasil NSP tersebut. Berikut ini merupakan model dari NSP :

1.4.1 Parameter

Berikut ini merupakan parameter-parameter yang digunakan :

n = Jumlah hari yang dijadwalkan

m = Jumlah perawat yang tersedia

i = Indeks untuk hari

k = Indeks untuk perawat

P_i = Kebutuhan jumlah perawat pada *shift* pagi

T_i = Kebutuhan jumlah perawat pada *shift* sore

M_i = Kebutuhan jumlah perawat pada *shift* malam

1.4.2 Variabel Keputusan

Berikut ini merupakan beberapa variabel keputusan yang digunakan dalam model NSP ini :

$$X_{i,k} = \begin{cases} 1 & \text{jika perawat ditugaskan pada } shift \text{ pagi pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$Y_{i,k} = \begin{cases} 1 & \text{jika perawat ditugaskan pada } shift \text{ sore pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$Z_{i,k} = \begin{cases} 1 & \text{jika perawat ditugaskan pada } shift \text{ malam pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$C_{i,k} = \begin{cases} 1 & \text{jika perawat tidak ditugaskan pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

1.4.3 Hard Constraint

Berikut ini merupakan formulasi dari *hard constraint* :

1. Jumlah minimal perawat di tiap *shift* harus dipenuhi

$$\sum_{k=1}^m X_{i,k} \geq P_i, i = 1, 2, \dots, n$$

$$\sum_{k=1}^m Y_{i,k} \geq T_i, i = 1, 2, \dots, n$$

$$\sum_{k=1}^m Z_{i,k} \geq M_i, i = 1, 2, \dots, n$$

2. Tiap perawat hanya mendapatkan 1 kali *shift* tiap hari

$$X_{i,k} + Y_{i,k} + Z_{i,k} = 1, i = 1, 2, \dots, n \text{ dan } k = 1, 2, \dots, m$$

3. Menghindari terdapat bentuk *shift* kerja “on-off-on”

$$C_{i,k} + X_{i,k} + Y_{i+1,k} + Z_{i+1,k} + C_{i+2,k} \leq 2, i = 1, 2, \dots, n-2 \text{ dan } k = 1, 2, \dots, m$$

4. Tiap perawat memiliki aturan mendapatkan *shift* malam 3 hari berturut-turut lalu diikuti 3 hari libur berturut-turut

$$Z_{1,k} + Z_{2,k} + Z_{3,k} + C_{4,k} + C_{5,k} + C_{6,k} + C_{19,k} + Z_{20,k} + Z_{21,k} = 9, k = 1, 7, 13$$

$$C_{1,k} + C_{2,k} + C_{3,k} + Z_{16,k} + Z_{17,k} + Z_{18,k} + C_{19,k} + Z_{20,k} + Z_{21,k} = 9, k = 2, 8, 14$$

$$Z_{13,k} + Z_{14,k} + Z_{15,k} + C_{16,k} + C_{17,k} + C_{18,k} = 6, k = 3, 9, 15$$

$$Z_{10,k} + Z_{11,k} + Z_{12,k} + C_{13,k} + C_{14,k} + C_{15,k} = 6, k = 4, 10, 16$$

$$Z_{7,k} + Z_{8,k} + Z_{9,k} + C_{10,k} + C_{11,k} + C_{12,k} = 6, k = 5, 11, 17$$

$$Z_{4,k} + Z_{5,k} + Z_{6,k} + C_{7,k} + C_{8,k} + C_{9,k} = 6, k = 6, 12, 18$$

5. Tiap perawat memiliki *shift* kerja antara 12 sampai 14 hari per penjadwalan

$$\sum_{i=1}^n (X_{i,k} + Y_{i,k} + Z_{i,k}) \geq 12, k = 1, 2, \dots, n$$

$$\sum_{i=1}^n (X_{i,k} + Y_{i,k} + Z_{i,k}) \leq 14, k = 1, 2, \dots, n$$

6. Tiap perawat tidak boleh memiliki *shift* kerja selama 6 hari berturut-turut

$$C_{i,k} + C_{i+1,k} + C_{i+2,k} + C_{i+3,k} + C_{i+4,k} + C_{i+5,k} + C_{i+5,k} \geq 1$$

$$i = 1, 2, \dots, n-4 \text{ dan } k = 1, 2, \dots, m$$

$$\sum_{i=n-v}^n C_{i,k} + \sum_{i=1}^{6-v} C_{i,k+1} \geq 1, \quad v = 0, 1, \dots, 5; k = 1, 2, \dots, m-1$$

$$\sum_{i=n-v}^n C_{i,m} + \sum_{i=1}^{6-v} C_{i,1} \geq 1, \quad v = 0, 1, \dots, 5$$

7. Tiap perawat memiliki *shift* sore minimal 25% dari total *workload*

$$\sum_{i=1}^n Y_{i,k} \geq 3, k = 1, 2, \dots, m$$

8. Tiap perawat memiliki *shift* pagi minimal 30% dari total *workload*

$$\sum_{i=1}^n X_{i,k} \geq 4, k = 1, 2, \dots, m$$

1.4.4 Soft Constraint

Berikut ini merupakan formulasi dari *soft constraint*

1. Menghindari setelah *shift* sore diikuti dengan *shift* pagi atau *shift* malam di hari berikutnya

$$Y_{i,k} + X_{i+1,k} + Z_{i+1,k} \leq 1 \quad i = 1, 2, 3 \dots n-1 \text{ dan } k = 1, 2, 3, 4, \dots, m$$

$$Y_{n,k} + X_{1,k+1} + Z_{1,k+1} \leq 1 \quad k = 1, 2, 3, 4, \dots, m-1$$

$$Y_{n,m} + X_{1,1} + Z_{1,1} \leq 1$$

2. Menghindari setelah *shift* pagi diikuti dengan *shift* sore atau *shift* malam dihari berikutnya

$$X_{i,k} + Y_{i+1,k} + Z_{i+1,k} \leq 1 \quad i = 1, 2, 3 \dots n-1 \text{ dan } k = 1, 2, 3, 4, \dots, m$$

$$X_{n,k} + Y_{1,k+1} + Z_{1,k+1} \leq 1 \quad k = 1, 2, 3, 4, \dots, m-1$$

$$X_{n,m} + Y_{1,1} + Z_{1,1} \leq 1$$

3. Tiap perawat setidaknya mendapatkan 1 hari libur tiap minggu

$$C_{7,k} + C_{14,k} + C_{21,k} \geq 1 \quad k = 1, 2, 3, 4, \dots, m$$

4. Tiap perawat memiliki total *workload* yang sama

$$\sum_{i=1}^n (X_{i,k} + Y_{i,k} + Z_{i,k}) = 13, \quad k = 1, 2, 3 \dots, m$$

1.4.5 Goal

Berikut ini merupakan *goal* dari *soft constraint* yang ada

Goal1 : Menghindari setelah *shift* sore diikuti dengan *shift* pagi atau *shift* malam di hari berikutnya

$$Y_{i,k} + X_{i+1,k} + Z_{i+1,k} + n1_{i,k} - p1_{i,k} = 1, i = 1, 2, 3 \dots n - 1$$

$$k = 1, 2, 3, 4, \dots m$$

$$Y_{n,k} + X_{1,k+1} + Z_{1,k+1} + n1_{i,k} - p1_{i,k} = 1 \quad k = 1, 2, 3, 4, \dots m - 1$$

$$Y_{n,m} + X_{1,1} + Z_{1,1} + n1_{i,k} - p1_{i,k} = 1$$

Goal 2 : Menghindari setelah *shift* pagi diikuti *shift* sore atau *shift* malam di hari berikutnya

$$X_{i,k} + Y_{i+1,k} + Z_{i+1,k} + n2_{i,k} - p2_{i,k} = 1 \quad i = 1, 2, 3 \dots n - 1$$

$$k = 1, 2, 3, 4, \dots m$$

$$X + Y_{1,k+1} + Z_{1,k+1} + n2_{i,k} - p2_{i,k} = 1 \quad k = 1, 2, 3, 4, \dots m - 1$$

$$X_{n,m} + Y_{1,1} + Z_{1,1} + n2_{i,k} - p2_{i,k} = 1$$

Goal 3 : Tiap perawat setidaknya mendapatkan 1 hari libur tiap minggu

$$C_{7,k} + C_{14,k} + C_{21,k} + n3_{i,k} = 1 \quad k = 1, 2, 3, 4, \dots m$$

Goal 4 : Tiap perawat memiliki total *workload* yang sama

$$\sum_{i=1}^n (X_{i,k} + Y_{i,k} + Z_{i,k}) + n4_{i,k} - p4_{i,k} = 13, \quad k = 1, 2, 3 \dots m$$

1.4.6 Fungsi Tujuan

Berikut ini merupakan fungsi tujuan dari *goal* diatas

$$\min = \sum_{i=1}^n \sum_{k=1}^m p1_{i,k} + \sum_{i=1}^n \sum_{k=1}^m p2_{i,k} + \sum_{k=1}^m n3_k \sum_{k=1}^m (n4_k + p4_k)$$

Dimana fungsi tujuan ini adalah meminimasi deviasi yang terdapat pada tiap *soft constraint*, yaitu menghindari setelah *shift* sore diikuti dengan *shift* pagi atau *shift* malam di hari berikutnya, menghindari setelah *shift* pagi diikuti dengan *shift* sore atau *shift* malam dihari berikutnya, tiap perawat mendapatkan 1 hari libur tiap minggu, dan tiap perawat memiliki total *workload* yang sama.

1.5 Simulated Annealing

Dalam buku Santosa dan Willy (2011), dijelaskan bahwa Simulated Annealing (SA) adalah algoritma metaheuristik yang meniru perilaku fisik proses pendinginan baja yang mendidih secara perlahan. Jadi teknik ini meniru perilaku proses pemanasan baja yang dipanaskan sampai suhu tertentu yang kemudian baja tersebut dilakukan pendinginan secara perlahan. Ketika baja dipanaskan hingga suhu yang mendidih maka atom-atom pada baja akan bergerak bebas, lalu seiring dilakukannya pendinginan pada baja tersebut atom-atom yang pada awalnya bergerak bebas akan terbatas gerakannya akibat perubahan suhu tersebut. Ketika suhu turun dan atom-atom tidak dapat bergerak bebas, maka susunan dari atom tersebut menjadi lebih teratur dan dapat membentuk kristal yang mempunyai energi internal yang minimum ketika suhu baja turun.

Tetapi hasil dari pendinginan baja ini tidak selalu menghasilkan kristal-kristal yang bagus yang memiliki energi internal. Hal ini dikarenakan hasil dari kristal tersebut dipengaruhi oleh laju dari penurunan suhu saat melakukan pendinginan baja. Ketika laju penurunan suhu terlalu besar maka akan terbentuk *polycrystalline* dimana nilai energi internalnya lebih besar daripada kristal. Oleh karena itu suhu baja yang sangat panas dan mendidih ketika selesai dipanaskan harus diturunkan secara perlahan-lahan suhu tersebut agar mendapatkan status kristal yang memiliki energi internal yang minimum dan susunan struktur dari kristal yang bagus dapat terbentuk.

Proses pendinginan baja inilah yang dianalogikan sehingga diimitasi dalam metode *metaheuristik* dengan cara menentukan parameter suhu lalu mengontrolnya dengan menggunakan konsep distribusi probabilitas *Boltzmann*. Distribusi probabilitas *Boltzmann* menyatakan bahwa energi dari suatu sistem dalam keseimbangan panas pada suhu T terdistribusi secara probabilistik mengikuti rumus

$$P(E) = e^{-E/kT}$$

dimana

$P(E)$ = peluang mencapai tingkat energi E

T = suhu

K = konstanta

Metode yang mengimplementasikan distribusi probabilitas *Boltzmann* juga dapat diterapkan dalam minimasi fungsi, misalkan solusi yang sekarang adalah x dan nilai fungsinya $f(x)$, energi E_i pada status X_i adalah

$$E_i = f_i = f(x_i)$$

Menurut kriteria metropolis, probabilitas titik solusi berikutnya adalah x_{i+1} bergantung pada perbedaan status energi atau fungsi tujuan di dua titik diberikan oleh

$$P[E_{i+1}] = \min \{1, e^{-E/kT}\}$$

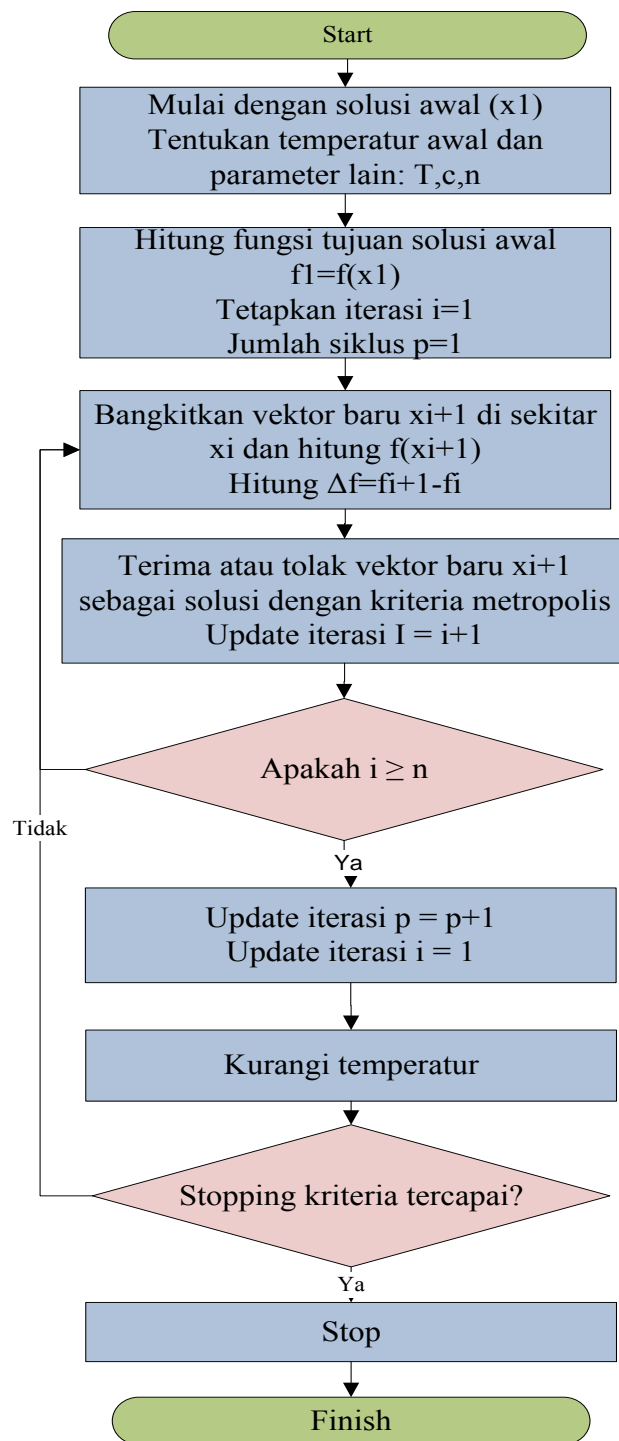
Dimana :

$$\Delta E = E_{i+1} - E_i = \Delta f = f_{i+1} - f_i = f(x_{i+1}) - f(x_i)$$

Titik baru bisa ditemukan dengan menggunakan distribusi probabilitas *Boltzmann*. Untuk faktor k diberikan nilai 1. Jika $\Delta E \leq 0$ maka $P[E_{i+1}] = 1$ sehingga titik x_{i+1} selalu diterima. Tetapi jika $\Delta E > 0$ maka nilai $f(x_{i+1})$ lebih besar dari $f(x)$. Tetapi x_{i+1} belum tentu ditolak dengan probabilitas P

$$P[E_{i+1}] = e^{-E/kT}$$

Secara umum algoritma *simulated annealing* digambarkan pada flowchart berikut ini



Gambar 2. 1 *flowchart simulated annealing*

1.6 Literature Review

Pada subbab ini akan akan dijelaskan posisi penelitian ini terhadap penelitian-penelitian sebelumnya mengenai *nurse scheduling problem* yang akan menjadi

dasar dalam penelitian ini. Berikut ini merupakan tabel dari penelitian-penelitian sebelumnya

Tabel 2.1 *Literature Review*

No	Penelitian	Metode	Tujuan	Ringkasan
1	(Jenal, Ruzakkiah et al., 2011) <i>A cyclical nurse schedule using goal programming</i>	<i>Goal programming</i>	Minimasi <i>goal</i>	Pada jurnal ini dalam melakukan penyelesaian penjadwalan perawat digunakan metode <i>goal programming</i> dimana tujuannya adalah meminimalkan pelanggaran yang terjadi di dalam penjadwalan tersebut
2	(Dowsland and thompson ., 2000) <i>solving a nurse scheduling problem with knapsacks, networks, and tabu search</i>	<i>tabu search</i>	Minimasi biaya	Dilakukan penyelesaian penjadwalan dimana pelanggaran yang terjadi akan dikalikan dengan biaya yang harus dikeluarkan oleh rumah sakit
3	(Bai et al,) <i>an efficient hybrid algorithm for nurse rostering problem</i>	<i>Hybrid algorithm</i>	Minimasi biaya	Dilakukan penyelesaian penjadwalan dimana pelanggaran yang terjadi akan dikalikan dengan

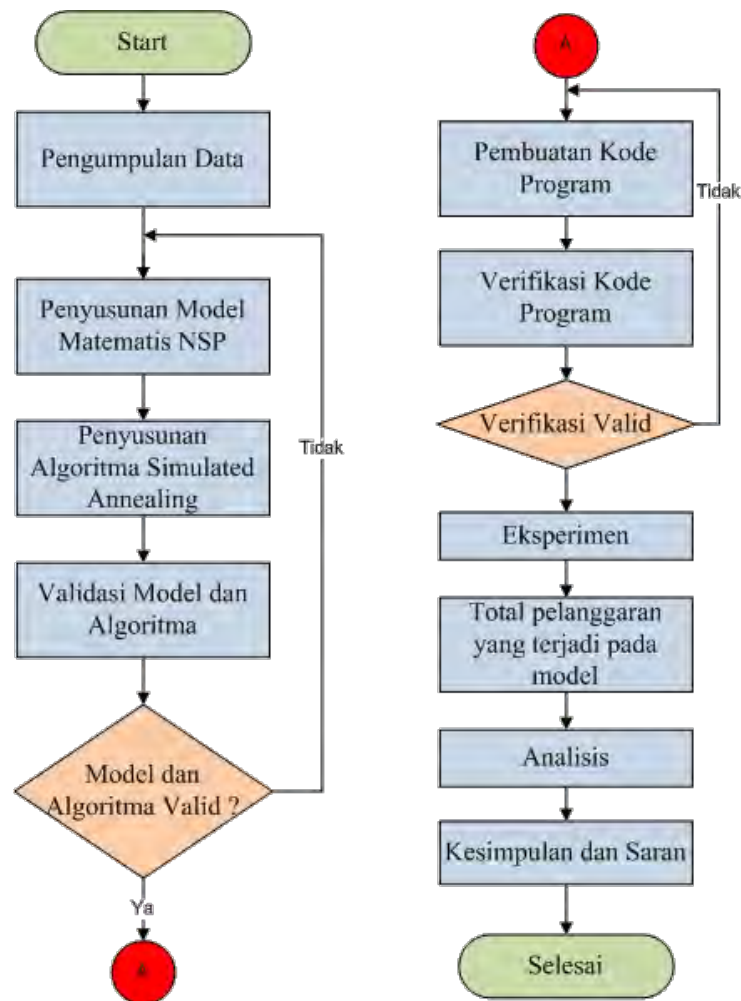
				biaya yang harus dikeluarkan oleh rumah sakit
4	(Ismail, WR,2012) <i>goal programming based master plan for cyclical nurse scheduling</i>	<i>Goal programming</i>	Minimasi <i>goal</i>	Dilakukan penyelesaian penjadwalan dengan metode <i>goal programming</i> dengan tujuannya adalah minimasi pelanggaran yang terjadi di hasil penjadwalan

Dari penelitian mengenai *nurse scheduling problem* peneliti memiliki tujuan untuk meminimasi pelanggaran yang ada dari *soft constraint* pada model matematis *nurse scheduling problem* dengan menggunakan algoritma *simulated annealing*. Pada penelitian sebelumnya algoritma yang digunakan dalam meminimasi pelanggaran dengan menggunakan metode *goal programming* yang akan membutuhkan waktu yang sangat lama jika menyelesaikan kasus yang besar. Sehingga peneliti menggunakan algoritma *simulated annealing* untuk menyelesaikan *nurse scheduling problem* dengan meminimasi pelanggaran yang terjadi.

BAB 3

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai tahapan-tahapan yang dilakukan dalam penelitian implementasi algoritma *simulated annealing* pada *nurse scheduling problem* (NSP) untuk minimasi pelanggaran yang terjadi pada model NSP. Berikut ini merupakan *flowchart* dari penelitian yang dilakukan



Gambar 3. 1 *Flowchart* penelitian

1.1 Pengumpulan Data

Pada subbab ini peneliti melakukan pengumpulan data penjadwalan perawat di RSUD Dr. Soetomo Surabaya dimana hasil dari pengumpulan data ini akan digunakan dalam penyusunan model matematis dalam *nurse scheduling problem*. Data yang didapatkan yaitu *shift* kerja perawat, jumlah perawat tiap *shift*, *workload* perawat selama 1 bulan, dan aturan-aturan khusus yang sesuai dengan peraturan di rumah sakit. Setelah mendapatkan data-data tersebut, maka selanjutnya akan dilakukan pembuatan model matematis yang akan disesuaikan dengan kondisi *real* penjadwalan perawat di RSUD Dr. Soetomo Surabaya.

1.2 Penyusunan Model Matematis Nurse Scheduling Problem

Setelah mendapatkan data penjadwalan perawat di RSUD Dr. Soetomo Surabaya, selanjutnya peneliti menyusun model matematis yang sesuai dengan kondisi *real* di rumah sakit tersebut. Berikut ini merupakan model matematis dari *nurse scheduling problem* di RSUD Dr. Soetomo Surabaya

1.2.1 Parameter

Berikut ini merupakan parameter-parameter yang digunakan :

n =Jumlah hari penjadwalan

m =Jumlah perawat yang tersedia

i =Indeks hari

j =Indeks perawat

a = Total *workload shift* pagi tiap perawat

b = Total *workload shift* sore tiap perawat

c = Total *workload shift* malam tiap perawat,

d = Jumlah libur perawat

p_i = Kebutuhan minimal perawat pada *shift* pagi pada hari ke i

q_i = Kebutuhan minimal perawat pada *shift* sore pada hari ke i

r_i = Kebutuhan minimal perawat pada *shift* malam pada hari ke i

1.2.2 Variabel Keputusan

Berikut ini merupakan variabel keputusan yang digunakan dalam model NSP ini :

$$Xp_{i,j} = \begin{cases} 1 & \text{jika perawat ditugaskan pada shift pagi pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$Xq_{i,j} = \begin{cases} 1 & \text{jika perawat ditugaskan pada shift sore pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$Xr_{i,j} = \begin{cases} 1 & \text{jika perawat ditugaskan pada shift malam pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

$$Xo_{i,j} = \begin{cases} 1 & \text{jika perawat tidak ditugaskan pada hari ke } i \\ 0 & \text{jika tidak} \end{cases}$$

1.2.3 Hard Constraint

Berikut ini merupakan formulasi dari *hard constraint* :

1. Jumlah minimal perawat tiap *shift* harus dipenuhi

$$\sum_{j=1}^m Xp_{i,j} \geq P_i, i = 1, 2, \dots, n$$

$$\sum_{j=1}^m Xq_{i,j} \geq T_i, i = 1, 2, \dots, n$$

$$\sum_{j=1}^m Xr_{i,j} \geq M_i, i = 1, 2, \dots, n$$

2. Tiap perawat hanya mendapatkan 1 kali *shift* tiap hari

$$Xp_{i,j} + Xq_{i,j} + Xr_{i,j} = 1, i = 1, 2, \dots, n \text{ dan } j = 1, 2, \dots, m$$

3. *Consecutive day* : dimana tiap perawat mendapatkan 2 *shift* malam berturut-turut dan diikuti 1 hari libur

$$Xr_{i,j} \leq Xr_{i+1,j}$$

$$1 - Xr_{i,j} \leq 9999 * (1 - Xr_{i+1,j})$$

$$Xr_{i,j} \leq Xo_{i+2,j}$$

$$Xr_{i,j} + Xr_{i+1,j} = 2 * Xo_{i+2,j}$$

Dimana $i=1,2,3,\dots,n-3$ dan $j=1,2,3,\dots,m$

$$Xr_{i,j} \leq Xr_{i+1,j}$$

$$1 - Xr_{i,j} \leq 9999 * (1 - Xr_{i+1,j})$$

Dimana $i=29$ dan $j=1,2,3,\dots,m$

1.2.4 *Soft Constraint*

Berikut ini merupakan formulasi dari *soft constraint*

1. *Workload* perawat per *shift* selama 1 bulan

$$\sum_{i=1}^n xp_{i,j} = a \quad j = 1,2,3 \dots m$$

$$\sum_{i=1}^n xq_{i,j} = b \quad j = 1,2,3 \dots m$$

$$\sum_{i=1}^n xr_{i,j} = c \quad j = 1,2,3 \dots m$$

2. Jumlah libur perawat tiap bulan

$$\sum_{i=1}^n xo_{i,j} = d \quad j = 1,2,3 \dots m$$

3. Menghindari *shift* kerja “*off-on-off*”

$$xo_{i,j} + xp_{i+1,j} + xq_{i+1,j} + xr_{i+1,j} + xo_{i+2,j} \leq 2$$

Dimana $i=1,2,3 \dots n-2$ dan $j=1,2,3 \dots m$

4. Menghindari setelah *shift* sore diikuti *shift* pagi

$$xq_{i,j} + xp_{i+1,j} \leq 1 \quad i = 1,2,3 \dots n-1 \text{ dan } j = 1,2,3,4, \dots m$$

5. Menghindari setelah *shift* pagi diikuti *shift* malam

$$xp_{i,j} + xr_{i+1,j} \leq 1 \quad i = 1,2,3 \dots n-1 \text{ dan } j = 1,2,3,4, \dots m$$

1.2.5 *Goal*

Berikut ini merupakan *goal* dari *soft constraint*

1. *Workload* perawat per *shift* selama 1 bulan (3)

$$\sum_{i=1}^n xp_{i,j} + u1_{i,j} - e1_{i,j} = a \quad j = 1,2,3 \dots m$$

$$\sum_{i=1}^n xq_{i,j} + u2_{i,j} - e2_{i,j} = b \quad j = 1,2,3 \dots m$$

$$\sum_{i=1}^n xr_{i,j} + u3_{i,j} - e3_{i,j} = c \quad j = 1,2,3 \dots m$$

2. Jumlah libur perawat tiap bulan (3)

$$\sum_{i=1}^n x_{o_{i,j}} + u_{4_{i,j}} - e_{4_{i,j}} = d \quad j = 1, 2, 3 \dots m$$

3. Menghindari *shift* kerja “*off-on-off*” (2)

$$x_{o_{i,j}} + x_{p_{i+1,j}} + x_{q_{i+1,j}} + x_{r_{i+1,j}} + x_{o_{i+2,j}} + u_{5_{i,j}} - e_{5_{i,j}} = 2$$

Dimana $i=1, 2, 3 \dots n-2$ dan $j=1, 2, 3 \dots m$

4. Menghindari setelah *shift* sore diikuti *shift* pagi (4)

$$x_{q_{i,j}} + x_{p_{i+1,j}} + u_{6_{i,j}} - e_{6_{i,j}} = 1$$

$i = 1, 2, 3 \dots n - 1$ dan $j = 1, 2, 3, 4, \dots m$

5. Menghindari setelah *shift* pagi diikuti *shift* malam (2)

$$x_{p_{i,j}} + x_{r_{i+1,j}} + u_{7_{i,j}} - e_{7_{i,j}} = 1$$

$i = 1, 2, 3 \dots n - 1$ dan $j = 1, 2, 3, 4, \dots m$

1.2.6 Bobot Kepentingan Fungsi Tujuan

Pemberian bobot ini dilakukan untuk mendapatkan tingkat kepentingan dari fungsi dari *goal* dimana tidak semua fungsi *goal* ini mempunyai tingkat kepentingan yang sama sehingga untuk membedakan hal tersebut harus dilakukan pemberian bobot terhadap fungsi dari *goal* untuk mendapatkan fungsi tujuan yang sesuai. Berikut ini merupakan bobot yang digunakan pada fungsi *goal* :

1. $W_1 = 3$, yaitu bobot untuk meminimalkan deviasi *workload* perawat per *shift* tiap bulannya. Pemberian bobot sebesar 3 ini agar *workload* yang diterima tiap perawat ini sama, yaitu pada *shift* pagi, sore, dan malam sehingga perbedaan *workload* tiap perawat pun dapat di minimalkan.
2. $W_2 = 3$, yaitu bobot untuk meminimalkan deviasi jumlah libur perawat tiap bulannya. Pemberian bobot sebesar 3 ini agar *workload* yang diterima tiap perawat ini sama, yaitu pada *shift* libur sehingga perbedaan *workload* tiap perawat pun dapat di minimalkan.
3. $W_3 = 2$, yaitu bobot untuk meminimalkan deviasi perawat yang mempunyai *shift* kerja “*off-on-off*”. Pemberian bobot sebesar 2 agar tiap perawat memiliki pola kerja seperti itu dapat di minimalkan karena dapat menurunkan performansi dari perawat tersebut yaitu setelah libur akan bekerja lalu libur lagi di hari selanjutnya.
4. $W_4 = 4$, yaitu bobot untuk meminimalkan deviasi perawat yang ditugaskan setelah *shift* sore diikuti *shift* pagi. Pemberian bobot yang besar yaitu 4 agar perawat tidak mendapatkan pola tersebut,

dikarenakan perawat akan kelelahan secara fisik karena terus bekerja pada shift sore yaitu dimulai pukul 4 sore hingga 12 malam lalu dilanjutkan pada shift pagi dimulai pukul 8 pagi di hari berikutnya.

5. $W_5 = 2$, yaitu bobot untuk meminimalkan deviasi perawat yang ditugaskan setelah *shift* pagi diikuti *shift* malam. Pemberian bobot sebesar 2 ini agar perawat tidak mendapatkan pola bekerja seperti ini dikarenakan waktu istirahat yang didapatkan terlalu panjang sehingga akan menurunkan motivasi dari perawat untuk bekerja.

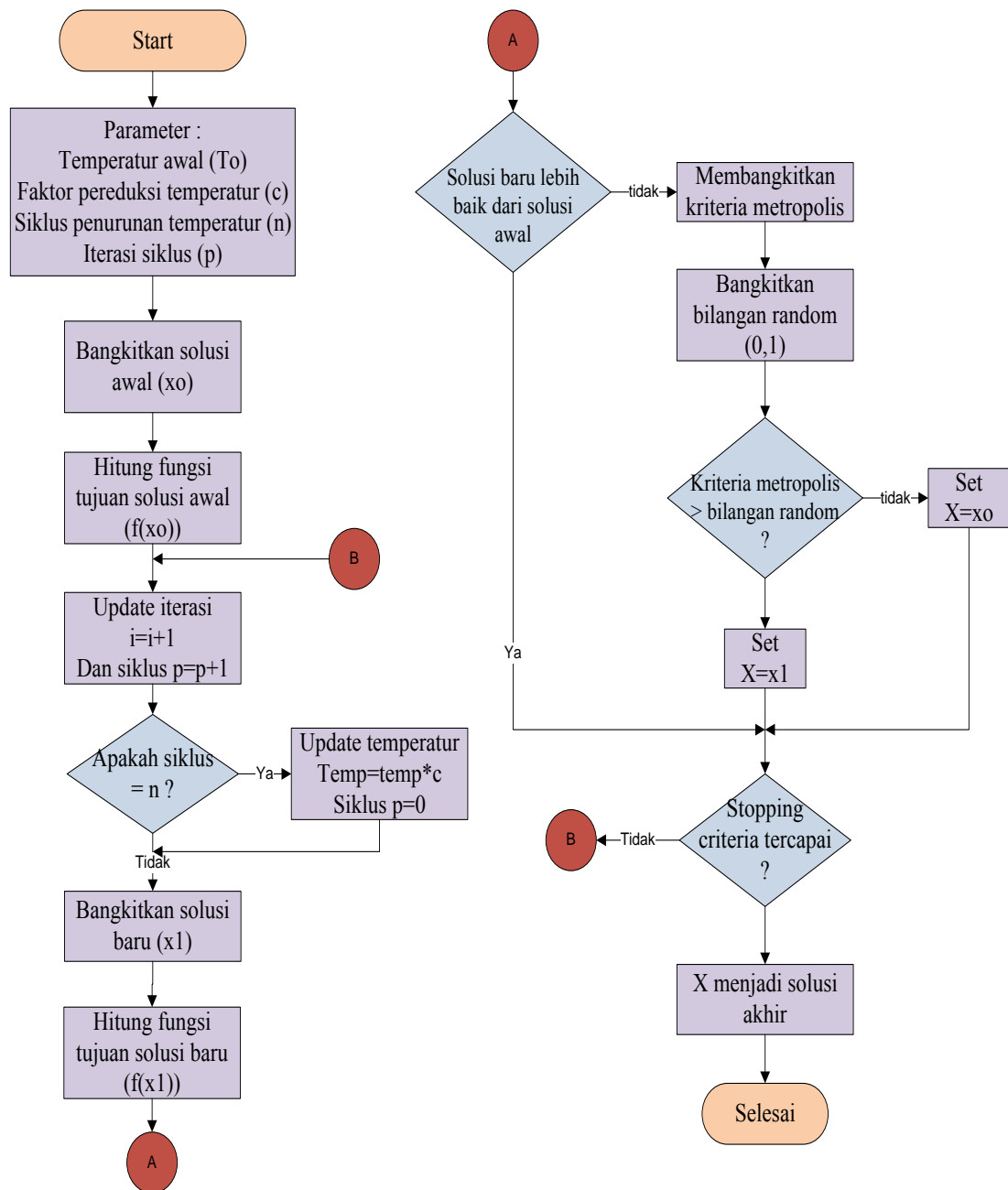
1.2.7 Fungsi Tujuan

Berikut ini merupakan fungsi tujuan dari *nurse scheduling problem*

$$\begin{aligned} \min = & w_1 * \sum_{i=1}^n \sum_{j=1}^m (u1_{i,k} + e1_{i,j}) + w_1 * \sum_{i=1}^n \sum_{j=1}^m (u2_{i,k} + e2_{i,j}) + \\ & w_1 * \sum_{i=1}^n \sum_{j=1}^m (u3_{i,k} + e3_{i,j}) + w_2 \sum_{i=1}^n \sum_{j=1}^m (u4_{i,k} + e4_{i,j}) + \\ & w_3 * \sum_{i=1}^n \sum_{k=1}^m e5_{i,k} + w_4 \sum_{i=1}^n \sum_{j=1}^m e6_{i,j} + w_5 * \sum_{i=1}^n \sum_{j=1}^m e7_{i,j} \end{aligned}$$

1.3 Penyusunan Algoritma *Simulated Annealing*

Pada subbab ini sebelum dilakukan penyusunan algoritma maka akan dilakukan pembuatan algoritma *simulated annealing* secara umum dimana algoritma *simulated annealing* ini merupakan metode utama yang akan digunakan dalam menyelesaikan masalah. Dalam melakukan penyusunan langkah-langkah jalannya metode *simulated annealing* akan disesuaikan dengan permasalahan pada penelitian ini, yaitu *nurse scheduling problem* dengan tujuan meminimasi pelanggaran yang terjadi pada *soft constraint*. Berikut ini merupakan algoritma dari *simulated annealing*



Gambar 3. 2 Algoritma *Simulated Annealing*

1.3.1 Penentuan Parameter

Tahap pertama dari algoritma *simulated annealing* adalah dengan menentukan parameter awal sebelum memulai algoritma tersebut. Hal ini dikarenakan penentuan parameter awal pada algoritma *simulated annealing* akan berpengaruh pada hasil akhir nantinya. Parameter-parameter yang terdapat pada algoritma *simulated annealing* adalah temperatur awal, faktor pereduksi

temperatur, iterasi siklus, dan iterasi. Temperatur awal merupakan nilai temperatur awal yang ditetapkan yaitu dimana biasanya temperatur awal dimulai dengan temperatur yang tinggi dan nanti temperatur itu akan turun bersamaan dengan bertambahnya iterasi. Faktor pereduksi temperatur merupakan faktor pereduksi temperatur setiap beberapa kali iterasi dalam 1 siklus bernilai 0 sampai 1. Siklus penurunan temperatur merupakan berisi dari beberapa kali iterasi sampai mencapai 1 siklus dan tiap mencapai 1 siklus tersebut akan dilakukan penurunan temperatur sebesar faktor pereduksi termperaturnya.

1.3.2 Bangkitkan solusi awal dan nilai fungsi tujuan solusi awal

Pada pembangkitan solusi awal ini dilakukan secara random yang sesuai dengan *constraint* yang sudah ditetapkan sebelumnya. Solusi awal ini merupakan solusi yang terbaik untuk sementara. Sebagai contoh pembangkitan solusi awal pada *nurse scheduling problem* sebagai berikut

Tabel 3. 1 Pembangkitan solusi awal

Perawat ke-	Hari							
	1	2	3	4	5	6	7
1	M	M	L	P	P	S	P	
2	S	S	M	M	L	P	S	
3	P	S	P	S	M	M	L	
.....								

Dimana P = *shift* pagi, S = *shift* sore, M = *shift* malam, L = Libur

Setelah dilakukan pembangkitan solusi awal secara random seperti pada contoh tabel 3.1 dimana penetapan *shift* kerja pada perawat dilakukan secara random tetapi masih sesuai dengan *constraint* yang telah ditetapkan sebelumnya. Setelah itu selanjutnya dilakukan perhitungan fungsi tujuan dari solusi awal tersebut yang akan menjadi pertimbangan di nilai fungsi tujuan yang baru pada iterasi berikutnya.

1.3.3 Update iterasi dan siklus

Update iterasi ini akan terus dilakukan untuk mengetahui seberapa banyak pengulangan yang dilakukan hingga mendapatkan solusi yang terbaik dan iterasi

ini juga bisa digunakan sebagai *stopping criteria* dalam permasalahan ini. Lalu juga dilakukan *update* siklus dimana siklus ini digunakan sebagai waktu kapan temperatur dapat diturunkan sebesar faktor pereduksinya. Biasanya siklus ini terdiri dari beberapa kali iterasi agar mendapatkan 1 kali siklus.

1.3.4 Pembangkitan solusi baru dan nilai fungsi tujuan solusi baru

Pembangkitan solusi baru ini dilakukan secara random berdasarkan nilai dari solusi sebelumnya. Setelah mendapatkan solusi baru maka selanjutnya adalah akan menghitung nilai fungsi tujuan dari solusi baru ini yang dapat digunakan untuk membandingkan apakah nilai fungsi tujuan baru ini lebih baik daripada nilai fungsi tujuan sebelumnya.

1.3.5 Membandingkan nilai fungsi tujuan solusi baru dengan nilai fungsi tujuan solusi sebelumnya

Setelah didapatkan struktur solusi baru dan mendapatkan fungsi tujuannya maka selanjutnya adalah dilakukan perbandingan antara fungsi tujuan yang baru dengan fungsi tujuan yang sebelumnya. Terdapat dua hasil dari perbandingan ini antara lain :

1. Jika solusi baru lebih baik daripada solusi sebelumnya, maka solusi yang baru ini merupakan solusi yang terbaik untuk sementara ini
2. Jika solusi baru tidak lebih baik daripada solusi sebelumnya, maka akan masuk kriteria metropolis yang akan ditentukan berdasarkan probabilitas *boltzmann*. Hasil dari probabilitas *boltzmann* ini berikutnya akan dibandingkan dengan bilangan random 0 sampai 1 untuk menentukan apakah solusi yang baru akan diterima atau tidak. Jika bilangan random lebih dari probabilitas *boltzmann* maka solusi baru akan diterima untuk sementara waktu, tetapi sebaliknya maka solusi baru akan ditolak dan akan menerima solusi yang lama. Ini merupakan kelebihan dari algoritma *simulated annealing* dimana walaupun solusi baru tidak lebih baik dari solusi yang lama akan tetap diterima untuk sementara dengan membandingkan nilai random dan probabilitas *boltzmann*

1.3.6 Stopping Criteria

Dalam kriteria pemberhentian ini, terdapat beberapa cara yang dapat digunakan. Pertama adalah dengan mempertimbangkan nilai dari temperatur dimana jika temperatur sudah sangat kecil maka bisa digunakan sebagai acuan untuk memberhentikan pengulangan yang dilakukan. Kedua adalah dengan menetapkan jumlah iterasi sehingga jika pengulangan sudah mencapai maksimal iterasi maka pengulangan akan berhenti dan akan didapatkan solusi dari permasalahan tersebut. Lalu yang ketiga adalah dengan membandingkan antara solusi baru dengan solusi lama. Jika perbedaan antara solusi baru dengan yang lama tidak jauh secara signifikan atau perbedaannya sangat kecil maka bisa dilakukan pemberhentian dari algoritma tersebut dan akan mendapatkan solusi dari permasalahan tersebut.

1.4 Validasi Model dan Algoritma

Setelah dilakukan pembuatan model dan penyusunan algoritma, maka langkah selanjutnya adalah dilakukan validasi. Validasi ini merupakan proses pengecekan apakah model yang sudah dibuat sebelumnya sudah mempresentasikan model konseptual dari *nurse scheduling problem*. Validasi ini akan dilakukan dengan metode eksak menggunakan *software lingo* dengan menggunakan permasalahan yang sama dan lebih sederhana. Jika validasi ini berhasil maka model tersebut sudah sesuai dengan model konseptual dari permasalahan dan bisa masuk ketahapan selanjutnya.

1.5 Pembuatan Kode Program

Tahapan pembuatan kode program ini merupakan tahapan dari penelitian untuk menerjemahkan model dan algoritma *simulated annealing* ke dalam *software* komputer. Dalam menerjemahkan model dan algoritma *simulated annealing* akan disesuaikan dengan permasalahan pada penelitian ini, yaitu *nurse scheduling problem*. Kode program ini akan digunakan dalam melakukan eksperimen-eksperimen untuk mendapatkan hasil yang terbaik.

1.6 Verifikasi

Verifikasi merupakan tahapan untuk mengetahui apakah kode program yang sudah dibuat sudah sesuai dengan logika pada algoritma tersebut. Verifikasi ini

juga dapat mengetahui apakah kode program tersebut terdapat *error* sehingga kode program tersebut tidak dapat berjalan. Jika verifikasi sudah berhasil dimana logika algoritma sudah sesuai dengan kode program dan tidak terdapat *error* maka selanjutnya akan dilakukan eksperimen terhadap permasalahan tersebut untuk mendapatkan hasil yang terbaik.

1.7 Eksperimen

Setelah dilakukan verifikasi dimana kode program sudah sesuai dengan logika algoritma dan tidak terdapat *error* ketika kode program dijalankan maka tahap selanjutnya adalah dilakukan eksperimen menggunakan kode program tersebut. Eksperimen ini dilakukan untuk mendapatkan hasil yang terbaik dari *nurse scheduling problem*. Untuk mendapatkan hasil yang terbaik, pada eksperimen akan dilakukan perubahan-perubahan parameter-parameter awal pada algoritma *simulated annealing*, yaitu dilakukan perubahan pada temperatur awal, faktor pereduksi temperatur, maksimal iterasi dalam 1 siklus dan faktor pemberhentian lain. Selain itu juga dilakukan eksperimen dengan metode eksak. Hasil dari eksperimen ini akan di rekap dan dilihat hasil fungsi tujuan dari *nurse scheduling problem*, yaitu minimasi pelanggaran yang terjadi.

1.8 Analisis

Setelah dilakukan eksperimen maka selanjutnya dilakukan analisis terhadap hasil dari eksperimen tersebut. Analisis ini dilakukan terhadap hasil output dari eksperimen ini yaitu jumlah pelanggaran terjadi pada *nurse scheduling problem* dan juga waktu komputasi dari permasalahan ini. Analisis juga dapat dilakukan terhadap parameter-parameter awal yang diubah-ubah ketika dilakukannya eksperimen sehingga akan didapatkan rekomendasi parameter-parameter awal berapa saja yang terbaik agar mendapatkan hasil yang terbaik dan dengan waktu komputasi yang tidak lama jika akan dilakukan penelitian dengan kasus yang serupa.

1.9 Kesimpulan dan Saran

Setelah dilakukan analisis maka akan dilakukan penarikan kesimpulan dan saran dimana penarikan kesimpulan ini merupakan jawaban dari tujuan dari penelitian ini dilakukan. Selain itu kesimpulan juga berdasarkan hasil analisis dan

interpretasi hasil eksperimen yang sudah dilakukan. Setelah dilakukan penarikan kesimpulan maka selanjutnya diberikan saran mengenai rekomendasi-rekomendasi yang terbaik terhadap pengembangan dari penelitian ini yang bisa dilakukan pada penelitian selanjutnya.

BAB IV

PENGUJIAN ALGORITMA

Bab ini berisi tentang validasi dan verifikasi algoritma yang diusulkan sehingga algoritma tersebut dapat dipercaya untuk digunakan dalam penyelesaian implementasi algoritma *simulated annealing* pada *nurse scheduling problem* dengan studi kasus RSUD Dr. Soetomo Surabaya.

4.1 Verifikasi dan Validasi Model Matematis

Verifikasi model matematis dilakukan dengan cara mengevaluasi struktur model yang di *generate* di dalam *software* LINGO. Evaluasi model ini didasari apakah model yang dibuat di LINGO sudah merepresentasikan dengan model matematis. Jika setelah dievaluasi model yang di *generate* di LINGO sudah merepresentasikan dengan model matematisnya maka dapat disimpulkan jika verifikasi model matematis sudah benar. Lalu juga dilakukan validasi model menggunakan metode eksak, yaitu *branch and bound* dengan menggunakan bantuan *software* LINGO dimana dalam melakukan validasi model ini menggunakan contoh kasus kecil yaitu menggunakan 4 perawat dan 8 hari kerja. Dengan dilakukannya validasi model ini akan dilihat apakah model matematis yang digunakan bisa berjalan untuk dilakukan penyelesaian permasalahan ini. Berikut ini merupakan tabel data-data yang digunakan pada contoh kasus kecil yang digunakan dalam melakukan validasi algoritma

Tabel 4. 1 Data contoh kasus kecil

Data	Jumlah
Minimal perawat <i>shift</i> pagi	1
Minimal perawat <i>shift</i> sore	1
Minimal perawat <i>shift</i> malam	1
<i>shift</i> libur	1
<i>workload shift</i> pagi	2
<i>workload shift</i> sore	2
<i>workload shift</i> malam	2

Data pada tabel 4.1 diatas digunakan untuk mencari nilai optimal dalam *nurse scheduling problem* dari tujuannya yaitu minimasi deviasi terhadap aturan dari rumah sakit. Dengan menggunakan 4 perawat dan 8 hari kerja pada contoh

kasus kecil ini maka akan mendapatkan pola penjadwalan yang paling baik terhadap aturan-aturan yang berlaku pada rumah sakit. Berikut ini merupakan hasil dari dilakukannya validasi model pada contoh kasus tersebut.

Tabel 4. 2 Pola Penjadwalan

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4
1	p	s	p	m
2	s	l	p	m
3	m	p	s	l
4	m	p	s	p
5	l	s	m	p
6	p	l	m	s
7	p	m	l	s
8	s	m	p	s

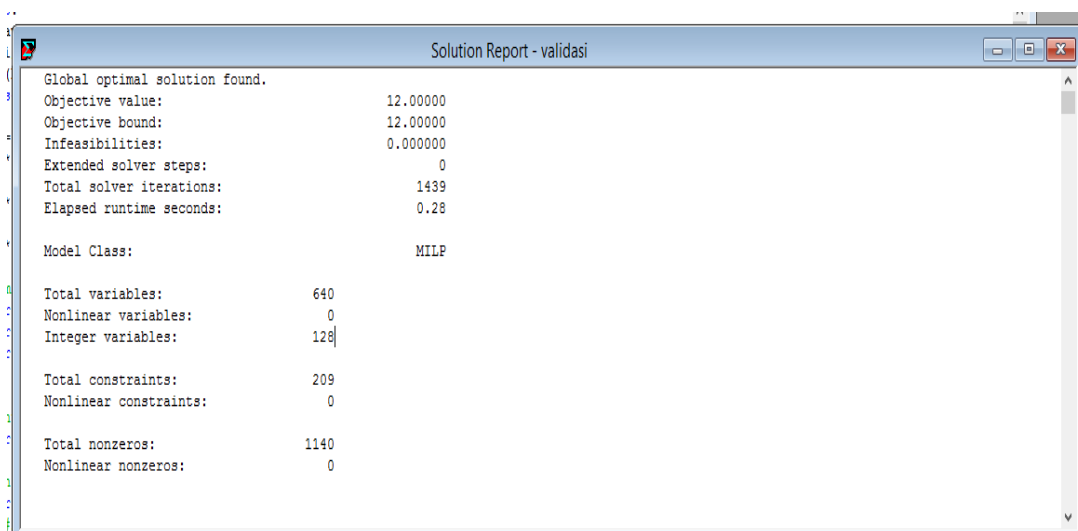
Keterangan :

P = *Shift* jaga pagi

S = *Shift* jaga sore

M = *Shift* jaga malam

L = Libur



Gambar 4. 1 Hasil Contoh Kasus Kecil Menggunakan Lingo

Dari hasil pola penjadwalan pada tabel 4.2 diatas maka didapatkan nilai Z sebesar 12 yang terdapat pada gambar 4.1. dikarenakan masih terdapat pelanggaran yang terjadi dari aturan yang ada berdasarkan hasil dari pola

penjadwalan tersebut. Berikut ini pemaparan dari hasil pola penjadwalan berdasarkan model matematis pada permasalahan *nurse scheduling problem*

A. Hard Constraint

Pada *hard constraint* ini dimana semua constraint ini harus dipenuhi dan tidak boleh dilanggar

1. Jumlah minimal perawat tiap *shift*

Dari contoh kasus kecil diatas menggunakan data dimana jumlah minimal perawat pada *shift* pagi, *shift* sore, dan *shift* malam adalah 1. Sehingga jika dilihat dari hasil pola penjadwalan yang sudah dilakukan dimana tiap harinya mulai hari ke-1 hingga hari ke-8 sudah sesuai dengan ketentuan. Sebagai contoh pada hari ke-1 didapatkan jika pada *shift* pagi terdapat perawat sebanyak 2, *shift* sore terdapat perawat sebanyak 1, dan *shift* malam terdapat perawat sebanyak 1.

2. Tiap perawat hanya mendapatkan 1 kali *shift* tiap hari

Jika dilihat dari hasil pola penjadwalan diatas didapatkan jika tiap perawat tiap harinya hanya mendapatkan 1 kali *shift* jaga.

3. Aturan khusus dimana tiap perawat mendapatkan 2 *shift* malam berturut-turut dan diikuti 1 hari libur

Aturan khusus berlaku jika terdapat perawat yang mendapatkan *shift* malam maka hari berikutnya perawat tersebut akan mendapatkan *shift* malam lagi lalu diikuti dengan 1 hari libur. Jika dilihat dari pola penjadwalan dimana pola *shift* tiap perawat sudah sesuai dengan aturan khusus tersebut. Sebagai contoh pada perawat 1 mendapatkan pola jika pada hari ke-3 dan ke-4 perawat tersebut mendapatkan *shift* malam, lalu pada hari ke-5 perawat tersebut libur. Tetapi terdapat pengecualian terhadap perawat 2 dikarenakan jumlah hari yang digunakan adalah 8 hari sementara pada perawat 2 ini mendapatkan *shift* malam pada hari ke-7 dan hari ke-8 sehingga untuk hari berikutnya yang harusnya mendapatkan libur akan diganti dengan hari-hari sebelumnya.

B. Soft constraint

Pada soft constraint ini dimana semua constraint dapat dilanggar tapi seminimal mungkin.

1. *Workload* perawat per *shift*

Workload perawat per *shift* pada contoh kasus kecil yaitu jumlah *shift* pagi, sore, dan malam tiap perawat masing-masing berjumlah 2 untuk selama 8 hari. Pada *constraint* ini terdapat pelanggaran yaitu pada perawat 1 mendapatkan *shift* pagi sebanyak 3, pada perawat 3 mendapatkan *shift* pagi sebanyak 3, lalu pada perawat 4 mendapatkan *shift* sore sebanyak 3. Dikarenakan terdapat pelanggaran maka pelanggaran yang terjadi akan dikalikan bobot pada *constraint* ini.

2. Jumlah libur perawat

Jumlah libur perawat pada contoh kasus kecil ini berjumlah 1. Tetapi pada hasilnya terdapat 1 pelanggaran yang terjadi, yaitu pada perawat 2 dengan libur berjumlah 2. Dikarenakan terdapat pelanggaran pada *constraint* ini maka pelanggaran yang terjadi dikalikan dengan bobot pada *constraint* ini.

3. Menghindari *shift* kerja “off-on-off”

Pada hasil contoh kasus kecil ini dapat dilihat tidak terjadi pelanggaran terhadap *constraint* ini yaitu perawat mendapatkan pola kerja libur kerja dan libur. Oleh karena itu pelanggaran yang terjadi bernilai 0.

4. Menghindari setelah *shift* sore diikuti *shift* pagi

Pada hasil contoh kasus kecil ini dapat dilihat tidak terjadi pelanggaran terhadap *constraint* ini dimana hasil pola penjadwalan tersebut tidak terjadi pola pada perawat setelah *shift* sore diikuti *shift* pagi.

5. Menghindari setelah *shift* pagi diikuti *shift* malam

Pada hasil contoh kasus kecil ini dapat dilihat tidak terjadi pelanggaran terhadap *constraint* ini dimana pola penjadwalan

tersebut tidak terjadi pola pada perawat dimana setelah *shift* pagi diikuti *shift* malam.

4.2 Langkah-Langkah dalam *Simulated Annealing*

Pada subbab ini akan dilakukan penjelasan mengenai langkah-langkah yang digunakan dalam algoritma *simulated annealing* dalam menyelesaikan *nurse scheduling problem* dengan menggunakan contoh kasus kecil yang sama ketika dilakukan validasi model pada subbab sebelumnya yaitu dengan menggunakan 4 perawat dan 8 hari. Berikut ini merupakan data yang digunakan dalam contoh kasus kecil

Tabel 4. 3 Data Contoh Kasus Kecil

Data	Jumlah
Minimal perawat <i>shift</i> pagi	1
Minimal perawat <i>shift</i> sore	1
Minimal perawat <i>shift</i> malam	1
<i>shift</i> libur	1
<i>workload shift</i> pagi	2
<i>workload shift</i> sore	2
<i>workload shift</i> malam	2

Langkah 1 : Inisialisasi Parameter

Parameter yang digunakan pada algoritma *simulated annealing* untuk menyelesaikan contoh kasus kecil adalah sebagai berikut :

- Temperatur awal = 8000
- Faktor pereduksi temperatur = 0.8
- Siklus penurunan temperatur = 5

Langkah 2 : Pembangkitan Solusi Awal

Hasil dari pembangkitan solusi awal sangat menentukan solusi akhir yang akan didapatkan. Jika solusi awal yang buruk maka akan membutuhkan iterasi dan waktu yang lebih lama untuk mendapatkan solusi akhir yang lebih baik. Pada contoh kasus kecil pada *nurse scheduling problem*, hasil dari solusi awal sangat beragam dikarenakan ada banyak kemungkinan yang terjadi pada pola penjadwalan yang ditentukan dikarenakan *shift* penjadwalan yang dilakukan dibangkitkan secara random. Sebagai contoh tiap perawat perharinya akan mendapatkan salah satu *shift* perharinya secara random sehingga hasil pola dari

penjadwalan akan semakin beragam, oleh karena itu pada kasus ini jika pada pembangkitan solusi awal mendapatkan hasil yang buruk maka solusi akhir yang didapatkan juga cenderung tidak bagus. Berikut ini cara yang digunakan dalam melakukan pembangkitan solusi awal :

- Jumlah perawat per *shift* tiap harinya

Pada penentuan jumlah perawat per *shift* di tiap harinya ini dilakukan dengan menggunakan rumus

```
p=ceil(rand*perawat);  
s=ceil(rand*(perawat-p));  
m=ceil(rand*(perawat-p-s));  
l=perawat-p-s-m;
```

Dimana p,s,m,l masing-masing adalah jumlah perawat pada *shift* pagi,sore,malam, dan libur di tiap harinya. Hasil dari penentuan jumlah perawat per *shift*nya harus sesuai dengan data pada contoh kasus kecil. Jika jumlah perawat per *shift* tersebut tidak sesuai dengan data maka akan *looping* hingga mendapatkan hasil yang sesuai dengan data.

- Pembangkitan solusi awal *shift* malam

Setelah didapatkan jumlah perawat per *shift* tiap harinya maka langkah selanjutnya adalah melakukan pembangkitan solusi awal secara random dimana yang pertama dilakukan pembangkitan solusi awal pada *shift* malam. Hal ini dikarenakan pada *shift* malam ini mempunyai pola yang unik jika dibandingkan dengan *shift* yang lain, yaitu pola *shift* malam lalu diikuti *shift* malam di hari berikutnya dan kemudian diikuti *shift* libur (malam-malam-libur) kecuali perawat yang mendapatkan pola pada 2 hari terakhir hanya mendapatkan pola *shift* malam lalu diikuti *shift* malam (malam-malam). Untuk mendapatkan hasil yang lebih baik maka pembangkitan solusi awal pada *shift* malam ini dilakukan pertama. Berikut merupakan contoh hasil pembangkitan solusi awal *shift* malam.

Tabel 4. 4 Solusi Awal *Shift* Malam

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4
1	-	-	-	m
2	-	-	-	m
3	-	m	-	l
4	-	m	-	-
5	-	l	m	-
6	-	-	m	-
7	m	-	l	-
8	m	-	-	-

Dari contoh hasil diatas didapatkan jika pada perawat 1 mendapatkan pola *shift* malam dimulai pada hari ke 7, perawat 2 mendapatkan pola *shift* malam dimulai pada hari ke 3, perawat 3 mendapatkan pola *shift* malam dimulai pada hari ke 5, dan perawat 4 mendapatkan pola *shift* malam dimulai pada hari ke 1

- Pembangkitan solusi awal

Setelah dilakukan pembangkitan solusi awal untuk *shift* malam, maka selanjutnya adalah dilakukan pembangkitan solusi awal secara keseluruhan untuk *shift-shift* yang lain secara random. Pembangkitan solusi awal ini dilakukan secara random dan sesuai dengan jumlah perawat yang dibutuhkan pada *shift* tersebut tiap harinya. Pembangkitan solusi awal secara random untuk *shift* yang lain,yaitu *shift* pagi, *shift* sore, dan *shift* malam dilakukan dengan cara :

- *Shift* pagi

Dalam mendapatkan *shift* pagi ini akan dibangkitkan bilangan random $r = \text{rand}$. Jika $r < 0.333$ maka perawat yang terpilih pada hari tersebut mendapatkan *shift* pagi, jika tidak maka akan masuk di kemungkinan yang lain yaitu *shift* sore atau libur.

- *Shift* sore

Dalam mendapatkan *shift* sore ini akan dibangkitkan bilangan random $r = \text{rand}$. Jika $0.333 < r < 0,667$ maka perawat yang terpilih pada hari tersebut mendapatkan *shift* sore, jika tidak

maka akan masuk di kemungkinan yang lain yaitu *shift* pagi atau libur.

➤ *Shift* libur

Dalam mendapatkan *shift* libur ini akan dibangkitkan bilangan random $r = \text{rand}$. Jika $r > 0.667$ maka perawat yang terpilih pada hari tersebut mendapatkan *shift* libur, jika tidak maka akan masuk di kemungkinan yang lain, yaitu *shift* pagi atau sore.

Berikut ini merupakan hasil dari pembangkitan solusi awal pada contoh kasus kecil :

Tabel 4. 5 Solusi Awal Contoh Kasus Kecil

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4
1	p	s	p	m
2	p	s	l	m
3	s	m	p	l
4	p	m	s	p
5	s	l	m	p
6	p	s	m	s
7	m	p	l	s
8	m	s	p	p

Pada tabel 4.5 didapatkan solusi awal dimana hasil dari solusi awal ini akan digunakan untuk mendapatkan solusi-solusi yang baru yang digunakan sebagai solusi akhir dalam penjadwalan perawat.

Langkah 3 Perhitungan Fungsi Tujuan solusi awal

Setelah didapatkan solusi awal maka selanjutnya adalah menghitung fungsi tujuan dari solusi awal yang sudah didapatkan dimana fungsi tujuannya adalah minimasi deviasi pelanggaran dari aturan yang berlaku di rumah sakit.

Tabel 4. 6 Solusi Awal dan Nilai Fungsi Tujuan dari Solusi Awal

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4	Z
1	p	s	p	m	54
2	p	s	l	m	
3	s	m	p	l	
4	p	m	s	p	
5	s	l	m	p	
6	p	s	m	s	
7	m	p	l	s	
8	m	s	p	p	

Dari tabel 4.6 diatas didapatkan jika nilai fungsi tujuan dari solusi awal tersebut sebesar 54.

Langkah 4 Penentuan Iterasi Awal dan Siklus Awal

Pada langkah ketiga ini setelah dilakukan penentuan solusi awal dan mendapatkan fungsi tujuan dari solusi awal tersebut maka selanjutnya adalah dilakukan penentuan iterasi awal dan siklus awal. Disini iterasi awal sama dengan 0 lalu siklus awal juga sama dengan 0 yang siklus ini digunakan sebagai penurunan temperatur.

Langkah 5 Pembangkitan Solusi Baru

Berikut ini merupakan langkah-langkah dalam melakukan pembangkitan solusi baru

1. Membangkitkan bilangan random untuk menentukan pada hari yang mana yang akan dilakukan pencarian solusi baru dengan menggunakan rumus :

$$ke = \text{ceil}(\text{hari} * \text{rand});$$

sebagai contoh jika nilai random yang didapatkan sebesar 0.3 , maka didapatkan hari yang akan dilakukan pencarian solusi baru yaitu pada hari ke 3.

2. Membangkitkan bilangan random untuk mendapatkan urutan dari penjadwalan yang baru

Untuk mendapatkan alternatif solusi baru maka bisa dilakukan dengan cara melakukan *flip*, *swap*, dan *slide*. Bilangan random yang dibangkitkan ini akan digunakan untuk menentukan

cara mana yang akan dipakai untuk digunakan pencarian solusi baru dengan kriteria sebagai berikut :

- Jika bilangan random yang dibangkitkan < 0.333 maka akan dilakukan *flip* untuk mendapatkan solusi baru.
- Jika bilangan random yang dibangkitkan diantara 0.333 dan 0.667 maka akan dilakukan *swap* untuk mendapatkan solusi baru.
- Jika bilangan random yang dibangkitkan > 0.667 maka akan dilakukan *slide* untuk mendapatkan solusi baru.

Misalkan bilangan random yang didapatkan adalah 0.5 maka akan dilakukan *swap* dalam mendapatkan solusi baru. Lalu akan dibangkitkan bilangan random sebanyak 2 yang digunakan sebagai perawat mana yang akan dilakukan *swap*. Misalkan hasilnya adalah perawat 1 dan perawat 3, maka *shift* perawat 1 pada hari ke 3 akan di *swap* dengan *shift* perawat 3 pada hari ke 3 sehingga akan didapatkan sebagai berikut :

Tabel 4. 7 Solusi Baru

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4
1	p	s	p	m
2	p	s	l	m
3	p	m	s	l
4	p	m	s	p
5	s	l	m	p
6	p	s	m	s
7	m	p	l	s
8	m	s	p	p

Langkah 6 Perhitungan Fungsi Tujuan Solusi Baru

Setelah didapatkan solusi baru maka selanjutnya adalah dilakukan perhitungan fungsi tujuan dari solusi baru tersebut dimana fungsi tujuannya adalah untuk meminimasi deviasi pelanggaran dari aturan yang berlaku di rumah sakit.

Tabel 4. 8 Solusi Baru dan Nilai Fungsi Tujuan Solusi Baru

Hari	Perawat 1	Perawat 2	Perawat 3	Perawat 4	Z
1	p	s	p	m	50
2	p	s	l	m	
3	p	m	s	l	
4	p	m	s	p	
5	s	l	m	p	
6	p	s	m	s	
7	m	p	l	s	
8	m	s	p	p	

Pada tabel 4.8 didapatkan jika nilai fungsi tujuan pada solusi baru sebesar 50.

Langkah 7 Membandingkan Solusi Baru dengan Solusi Lama

Setelah didapatkan solusi baru, maka solusi tersebut akan dibandingkan dengan solusi yang lama. Sesuai dengan algoritma *simulated annealing*, yaitu jika solusi baru lebih baik daripada solusi lama, maka solusi baru itu akan diterima, tetapi jika solusi baru tersebut tidak lebih baik daripada solusi lama maka akan masuk kriteria metropolis untuk menentukan apakah akan menerima solusi baru tersebut yang lebih buruk daripada solusi lama atau menolak solusi baru tersebut. Cara yang dilakukan adalah dengan membangkitkan bilangan random dan jika bilangan random tersebut lebih kecil daripada kriteria metropolis maka solusi baru yang tidak lebih barik dari solusi lama akan diterima, sebaliknya maka solusi baru akan ditolak.

Sebagai contoh jika nilai fungsi tujuan solusi lama sebesar 30, sedangkan nilai fungsi tujuan dari solusi baru sebesar 40, maka akan masuk kriteria metropolis dengan perhitungan sebagai berikut : (temperatur sekarang = 50)

$$\Delta E = Z_2 - Z_1 = 40 - 30 = 10$$

$$T = 50$$

$$P(E) = e^{-\Delta E/kT}$$

$$P(E) = e^{-10/50}$$

$$P(E) = 0.8187$$

Pada perhitungan diatas didapatkan jika nilai kriteria metropolis adalah 0.8187. Kemudian nilai kriteria metropolis itu akan dibandingkan dengan nilai bilangan random antara 0 sampai 1. Jika nilai bilangan random lebih kecil daripada nilai kriteria metropolis maka solusi baru tersebut akan diterima, tetapi jika nilai bilangan random lebih besar daripada nilai kriteria metropolis maka solusi baru tersebut akan ditolak. Sebagai contoh jika dibangkitkan nilai bilangan random antara 0 sampai 1 adalah $r = 0.5$. karena $r < P(E)$ maka solusi baru tersebut akan diterima walaupun nilai fungsi tujuan dari solusi baru tersebut tidak lebih baik daripada nilai fungsi tujuan solusi lama. Selanjutnya nilai solusi baru tersebut akan menjadi nilai dari solusi lama pada iterasi selanjutnya.

Langkah 8 *Stopping Criteria*

Setelah mendapatkan solusi baru maka langkah selanjutnya adalah *update* iterasi dan *update* siklus. Kemudian setelah didapatkan siklus sebesar S , maka akan dilakukan pengurangan temperatur sebesar Cr kemudian akan dilakukan pengecekan apakah temperatur sudah mencapai *stopping criteria*, yaitu 0.00000001. Jika temperatur sudah mencapai 0.00000001 maka akan dilakukan maka proses akan selesai dan mendapatkan solusi akhirnya.

Pada contoh kasus kecil ini setelah dilakukan penyelesaian dengan menggunakan algoritma *simulated annealing* maka didapatkan nilai fungsi tujuan yang sama dengan pengujian menggunakan metode eksak, yaitu *branch and bound* dengan menggunakan bantuan *software lingo* yaitu sebesar 12. Berikut ini merupakan hasil *output* menggunakan algoritma *simulated annealing*

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> [h,z,pl_pagi,pl_sore,pl_malam,pl_libur,pl_sorepagi,pl_pagimalam,pl_offonoff,iterasi]=sanspexkecil(4,8,8000,0.8)

h =

     1     1     4     1     2     2     3     3
     2     2     3     3     4     1     1     1
     1     4     2     2     3     3     4     1
     3     3     4     1     1     2     2     2

z =

    12

pl_pagi =

     6

pl_sore =

     3

pl_malam =

     0
```

Gambar 4. 2 Hasil *Output* Algoritma *Simulated Annealing* (1)

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

pl_malam =

     0

pl_libur =

     3

pl_sorepagi =

     0

pl_pagimalam =

     0

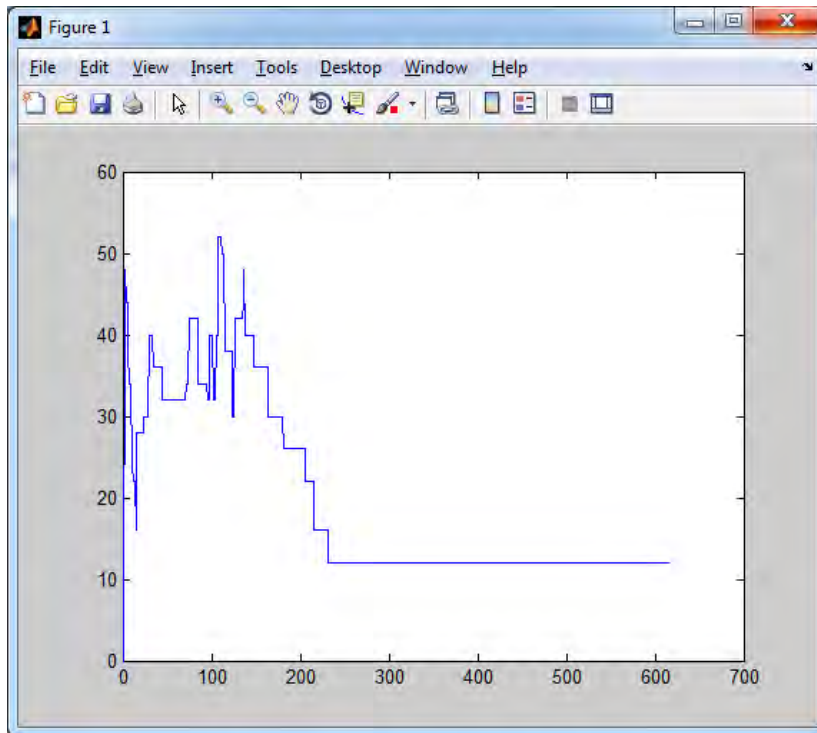
pl_offonoff =

     0

iterasi =

    616
```

Gambar 4. 3 Hasil *Output* Algoritma *Simulated Annealing* (2)



Gambar 4. 4 Grafik *Output* Penyelesaian dengan Algoritma *Simulated Annealing*

Dari hasil pada gambar 4.2 didapatkan jika nilai fungsi tujuan Z sebesar 12 dengan rincian dimana terdapat pelanggaran terhadap *workload* pada *shift* pagi sebesar 6, pelanggaran terhadap *workload* pada *shift* sore sebesar 3. Lalu pada gambar 4.3 didapatkan jika terdapat pelanggaran terhadap *workload* pada *shift* libur sebesar 3. Oleh karena itu didapatkan jika nilai fungsi tujuan Z pada solusi akhir ini sebesar 12. Lalu dapat dilihat pada gambar 4.4 dimana itu merupakan grafik nilai fungsi tujuan di tiap iterasi dimana hasilnya menjadi konvergen dan mendapatkan hasil yang optimal di akhir iterasi yang menandakan algoritma *simulated annealing* ini valid dan bisa digunakan untuk penyelesaian masalah yang lebih besar.

4.3 Verifikasi dan Validasi Algoritma

Verifikasi algoritma dilakukan untuk mengetahui apakah terdapat error dalam melakukan komputasi dengan *software* MATLAB. Verifikasi dilakukan

dengan mengecek setiap langkah logika perhitungan dari kode program yang dibuat apakah sudah sesuai dengan algoritma *simulated annealing*.

Validasi algoritma dilakukan dengan membandingkan hasil dari komputasi algoritma *simulated annealing* dengan hasil komputasi dengan menggunakan metode eksak. Dari subbab 4.2 dengan menggunakan contoh kasus kecil bisa dilihat output dari algoritma *simulated annealing* sama dengan output dari metode eksak sehingga dapat dikatakan jika algoritma yang dilakukan sudah valid.

BAB V

EKSPERIMEN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai hasil eksperimen yang telah dilakukan dan analisis terhadap eksperimen yang telah dilakukan.

1.1 Deskripsi Data Uji

Data yang digunakan dalam penyelesaian *nurse scheduling problem* berasal dari data di Rumah Sakit Dr. Soetomo Surabaya. Data uji ini akan digunakan sebagai inputan dalam melakukan penyelesaian *nurse scheduling problem* dan untuk mendapatkan nilai fungsi tujuan dari permasalahan ini. Berikut ini merupakan data yang digunakan dalam penyelesaian *nurse scheduling problem* sebagai berikut :

1. Jumlah perawat dan jumlah hari dalam penjadwalan

Jumlah perawat yang akan diselesaikan dalam permasalahan ini adalah berjumlah 85 orang yang berasal dari IRNA Medik dan dilakukan penjadwalan selama 30 hari.

2. Jumlah *shift* tiap perawat tiap harinya

Tiap perawat di tiap harinya akan mendapatkan 1 *shift* jaga, yaitu *shift* pagi, sore, malam, atau libur. Tidak diperbolehkan jika perawat mendapatkan lebih dari 1 *shift* tiap harinya.

3. Minimal jumlah perawat di tiap *shift* tiap harinya

Data ini digunakan untuk menjaga agar tiap harinya terdapat perawat yang harus berjaga pada *shift* tersebut dan tidak terjadi kekosongan *shift* jaga pada *shift* tertentu. Berikut ini merupakan rincian dari jumlah minimal perawat di tiap *shift* tiap harinya :

Tabel 5. 1 Rincian Kebutuhan Perawat Tiap *Shift* Tiap Harinya

Data	Jumlah (orang)
Minimal jumlah perawat <i>shift</i> pagi	25
Minimal jumlah perawat <i>shift</i> sore	17
Jumlah perawat <i>shift</i> malam	17

Dari tabel 5.1 diatas bisa dilihat jika jumlah kebutuhan perawat di *shift* pagi sebesar 25 perawat, jumlah kebutuhan perawat di *shift* sore sebesar 17 perawat, dan jumlah kebutuhan perawat di *shift* malam sebesar 17 perawat.

4. *Workload* perawat tiap shift

Data jumlah *workload* ini akan digunakan agar dapat menjadwalkan perawat sesuai dengan jumlah *workload* yang diterima atau paling tidak akan seminimal mungkin dari perbedaan jumlah *workload* untuk mendapatkan hasil penjadwalan yang baik. Berikut ini merupakan rincian jumlah *workload* yang diterima tiap perawat tiap shift :

Tabel 5. 2 Rincian *Workload* Tiap Perawat Tiap Shift

Data	Jumlah (shift)
Jumlah <i>workload shift</i> libur	7
Jumlah <i>workload shift</i> pagi	9
Jumlah <i>workload shift</i> sore	8
Jumlah <i>workload shift</i> malam	6

Dari tabel 5.2 diatas merupakan *workload* yang akan didapatkan tiap perawat selama 30 hari dimana tiap perawat mendapatkan jumlah shift libur sebanyak 7 kali, shift pagi sebanyak 9 kali, shift sore sebanyak 8 kali, dan shift malam sebanyak 6 kali.

1.2 Eksperimen

Eksperimen pada *nurse scheduling problem* menggunakan bantuan *software* MATLAB 7.10.0 (R210a) dan LINGO. *Software* MATLAB ini digunakan untuk menyelesaikan *nurse scheduling problem* dengan menggunakan algoritma *simulated annealing* sedangkan *software* LINGO digunakan untuk penyelesaian dengan menggunakan metode eksak. Eksperimen dilakukan sebanyak 3 kali dengan menggunakan data parameter yang berbeda untuk mendapatkan solusi akhir yang paling baik. Adapun parameter-parameter yang digunakan dalam *simulated annealing* adalah temperatur awal (T_0), Faktor pereduksi (cr), dan jumlah siklus penurunan temperatur (N). Spesifikasi komputer yang digunakan adalah dengan menggunakan Intel (R) Core (TM) i5-3210M 2.50 G.hz dengan RAM 4 GB.

1.2.1 Eksperimen dengan Algoritma *Simulated Annealing*

Pada subbab ini akan menjelaskan tentang eksperimen yang dilakukan dengan menggunakan algoritma *simulated annealing* pada *nurse scheduling problem*.

1.2.1.1 Uji Parameter Faktor Pereduksi Temperatur (Cr)

Uji parameter yang pertama ini dilakukan dengan mengubah parameter pada faktor pereduksi temperatur (Cr). Faktor pereduksi temperatur (Cr) yang digunakan dalam melakukan uji parameter ini adalah 0.5, 0.7, 0.9 dan 0.95 dengan temperatur tetap yaitu 500 dan jumlah siklus penurunan temperatur sebesar 5. Berikut ini hasil uji parameter yang dilakukan dengan melakukan perubahan pada faktor pereduksi temperatur (Cr) dengan replikasi sebanyak 5 kali :

Tabel 5. 3 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.5

To=500, N=5, Cr=0.5			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	181	3078	0,2808
2	181	3040	0,2808
3	181	3014	0,2496
4	181	2972	0,2496
5	181	3082	0,2496
Rata-Rata	181	3037,2	0,26208

Tabel 5. 4 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.7

To=500, N=5, Cr=0.7			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	351	2920	0,468
2	351	2920	0,468
3	351	2868	0,468
4	351	2900	0,468
5	351	2870	0,4524
Rata-Rata	351	2895,6	0,46488

Tabel 5. 5 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr)=0,9

To=500, N=5, Cr=0,9			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	1171	2550	1,4352

2	1171	2548	1,482
3	1171	2532	1,4352
4	1171	2528	1,4196
5	1171	2514	1,4352
Rata-Rata	1171	2534,4	1,44144

Tabel 5. 6 Deviasi Pelanggaran dengan Faktor Pereduksi Temperatur (Cr) = 0.95

To=500, N=5, Cr=0,95			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	2406	2242	2,886
2	2406	2286	2,8704
3	2406	2278	2,9016
4	2406	2328	2,8548
5	2406	2302	2,8704
Rata-Rata	2406	2287,2	2,87664

Dari hasil pengujian pada tabel 5.3, 5.4, 5.5, dan 5.6 dengan masing-masing sebanyak 5 kali replikasi didapatkan hasil pengujian dengan menggunakan faktor pereduksi temperatur yang mana yang paling baik. Dengan menggunakan Cr sebesar 0,5 maka didapatkan rata-rata deviasi pelanggaran sebesar 3037,2 dengan waktu komputasi 0,26208. Lalu dengan menggunakan Cr sebesar 0,7 didapatkan rata-rata deviasi pelanggaran sebesar 2895,6 dengan waktu komputasi 0,46488. Dengan menggunakan Cr sebesar 0,9 didapatkan rata-rata deviasi pelanggaran sebesar 2534,4 dengan waktu komputasi 1,44144. Dan dengan menggunakan Cr sebesar 0,95 didapatkan rata-rata deviasi pelanggaran sebesar 2287,2 dengan waktu komputasi 2,87644. Dari hasil pengujian dapat diketahui jika dengan menggunakan faktor pereduksi temperatur (Cr) sebesar 0.95 mendapatkan nilai deviasi pelanggaran yang paling minimal. Walaupun waktu komputasinya membutuhkan waktu yang lebih lama daripada hasil yang lain tapi kualitas solusi yang didapatkan jauh lebih baik daripada hasil dari faktor pereduksi temperatur (Cr) yang lain, yaitu 0,5, 0,7, dan 0,9. Oleh karena itu dalam penelitian ini akan digunakan faktor pereduksi temperatur sebesar 0,95.

1.2.1.2 Uji Parameter Siklus Penurunan Temperatur(N)

Setelah mendapatkan faktor pereduksi temperatur (Cr) sebesar 0.95 yang akan digunakan dalam penelitian ini, maka selanjutnya adalah dilakukan uji parameter dengan mengubah parameter pada siklus penurunan temperatur (N). Penentuan jumlah siklus penurunan temperatur (N) ini akan berpengaruh oleh hasil output yang akan didapat nantinya. Jumlah siklus penurunan temperatur (N) yang digunakan dalam eksperimen ini adalah 5, 10, dan 15. Berikut ini hasil uji parameter berdasarkan jumlah siklus penurunan temperatur (N) dengan temperatur awal (To) sebesar 500 dan dengan faktor pereduksi temperatur (Cr) = 0.95:

Tabel 5. 7 Deviasi Pelanggaran dengan Jumlah Siklus Penurunan Temperatur (N) = 10

To=500, Cr=0.95, N=10			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	4811	2164	5,8344
2	4811	2170	5,7252
3	4811	2174	5,7408
4	4811	2156	5,7096
5	4811	2196	5,7252
Rata-Rata	4811	2172	5,74704

Tabel 5. 8 Deviasi Pelanggaran dengan Jumlah Siklus Penurunan Temperatur (N) = 15

To=500, Cr=0.95, N=15			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	7216	1882	8,5645
2	7216	1864	8,6425
3	7216	1806	8,7205
4	7216	1872	8,5801
5	7216	1888	8,5645
Rata-Rata	7216	1862,4	8,61442

Dari tabel 5.6, 5.7 dan 5.8 dapat dilihat hasil dari uji parameter dengan jumlah siklus penurunan temperatur masing-masing sebesar 5, 10, dan 15 dengan replikasi sebanyak 5 kali. Dengan menggunakan N sebesar 5 maka didapatkan rata-rata deviasi pelanggaran sebesar 2287,2 dengan waktu komputasi 2,87664. Lalu dengan menggunakan N sebesar 10 maka didapatkan rata-rata deviasi

pelanggaran sebesar 2172 dengan waktu komputasi 5,74704. Dan jika menggunakan N sebesar 15 maka didapatkan rata-rata deviasi pelanggaran sebesar 1862,4 dengan waktu komputasi 8,61442. Dari hasil pengujian diatas dapat diketahui jika dengan menggunakan jumlah siklus penurunan temperatur (N) sebesar 15 maka mendapatkan nilai deviasi pelanggaran yang paling minimal. Walaupun membutuhkan waktu komputasi yang lebih panjang dan dengan jumlah iterasi yang lebih banyak tapi kualitas solusi yang didapat jauh lebih baik. Sehingga dalam penelitian ini akan digunakan jumlah siklus penurunan temperatur sebesar 15.

1.2.1.3 Uji Parameter Temperatur Awal (To)

Setelah mendapatkan faktor pereduksi temperatur (Cr) sebesar 0,95 dan siklus penurunan temperatur (N) sebesar 15 maka selanjutnya akan dilakukan uji parameter dengan mengubah parameter pada temperatur awal (To). Temperatur awal (To) yang digunakan dalam uji parameter ini adalah 2000, 5000, dan 15000. Berikut ini hasil eksperimen berdasarkan besar temperatur awal (To) yang digunakan dengan faktor pereduksi temperatur (Cr) sebesar 0,95 dan jumlah siklus penurunan temperatur (N) sebesar 15 :

Tabel 5. 9 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 2000

To=2000, Cr=0.95, N=15			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	7621	1902	9,204
2	7621	1834	9,0325
3	7621	1828	9,1261
4	7621	1800	9,0621
5	7621	1854	9,1261
Rata-Rata	7621	1843,6	9,11016

Tabel 5. 10 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 5000

To=5000, Cr=0.95, N=15			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	7891	1764	9,3601
2	7891	1776	9,4381
3	7891	1824	9,4537
4	7891	1800	9,3601

5	7891	1816	9,3757
Rata-Rata	7891	1796	9,39754

Tabel 5. 11 Deviasi Pelanggaran dengan temperatur awal (To) sebesar 15000

To=15000, Cr=0.95, N=15			
Replikasi Ke	Jumlah Iterasi	Deviasi Pelanggaran	Waktu Komputasi
1	8206	1738	9,8437
2	8206	1750	9,7345
3	8206	1718	9,7501
4	8206	1730	9,7657
5	8206	1754	9,8281
Rata-Rata	8206	1738	9,78442

Dari tabel 5.9, 5.10, dan 5.11 dapat dilihat hasil dari uji parameter dengan jumlah siklus penurunan temperatur masing-masing sebesar 2000, 5000, dan 15000 dengan replikasi sebanyak 5 kali. Dengan menggunakan To sebesar 2000 maka didapatkan rata-rata deviasi pelanggaran sebesar 1843,6 dengan waktu komputasi 9,11016. Lalu dengan menggunakan To sebesar 5000 maka didapatkan rata-rata deviasi pelanggaran sebesar 1796 dengan waktu komputasi 9,39754. Dan jika menggunakan To sebesar 15000 maka didapatkan rata-rata deviasi pelanggaran sebesar 1738 dengan waktu komputasi 9,78442. Dari hasil pengujian diatas dapat diketahui jika dengan menggunakan temperatur awal (To) sebesar 15000 maka mendapatkan nilai deviasi pelanggaran yang paling minimal. Walaupun membutuhkan waktu komputasi yang lebih panjang dan dengan jumlah iterasi yang lebih banyak tapi kualitas solusi yang didapat jauh lebih baik.

1.2.1.4 Eksperimen Algoritma *Simulated Annealing*

Setelah dilakukan uji parameter maka selanjutnya dilakukan eksperimen *nurse scheduling problem* dimana *stopping criteria* yang digunakan dalam eksperimen ini adalah minimal temperatur sebesar 10^{-6} . Berikut ini parameter-parameter awal yang digunakan dalam melakukan eksperimen :

1. Faktor pereduksi temperatur (Cr) sebesar 0,95
2. Jumlah siklus penurunan temperatur (N) sebesar 15
3. Temperatur awal (To) sebesar 15000

Berikut ini merupakan hasil yang didapat dari eksperimen algoritma *simulated annealing* pada *nurse scheduling problem* yang dilakukan :

Tabel 5. 12 Rincian Pelanggaran Uji Parameter dengan *Stopping Criteria* Temperatur sebesar 10^{-6}

Replikasi Ke-	Waktu (detik)	Jumlah Workload Shift Pagi	Jumlah Workload Shift Sore	Jumlah Workload Shift Malam	Jumlah Workload Shift Libur	Pola Sore-Pagi	Pola Pagi-Malam	Pola Off-On-Off	Total
1	9,8437	183	321	408	168	428	134	82	1724
2	9,7345	153	330	396	225	412	134	78	1728
3	9,7501	195	336	396	195	368	134	80	1704
4	9,7657	195	270	372	159	520	138	108	1762
5	9,8281	195	294	360	171	476	132	50	1678
6	9,7657	162	270	330	168	528	122	126	1706
7	9,6253	168	288	378	180	520	152	24	1710
8	9,6097	150	300	378	192	436	142	100	1698
9	9,6253	159	339	312	210	432	156	112	1720
10	9,7033	168	288	348	162	452	118	116	1652

Dari tabel diatas dapat dilihat hasil eksperimen algoritma *simulated annealing* berupa rincian deviasi pelanggaran yang terjadi di tiap replikasinya. Dari 10 replikasi yang dilakukan, dihasilkan jumlah deviasi pelanggaran yang paling minimal, yaitu pada replikasi ke 10 yaitu sebesar 1652.

Selain dilakukannya eksperimen dengan *stopping criteria* menggunakan minimal temperatur sebesar 10^{-6} maka selanjutnya akan dilakukan eksperimen dengan *stopping criteria* minimal jumlah iterasi sebesar 50.000. Berikut ini merupakan hasil dari eksperimen yang dilakukan :

Tabel 5. 13 Rincian Pelanggaran Uji Parameter dengan *Stopping Criteria* Jumlah Iterasi sebesar 50.000

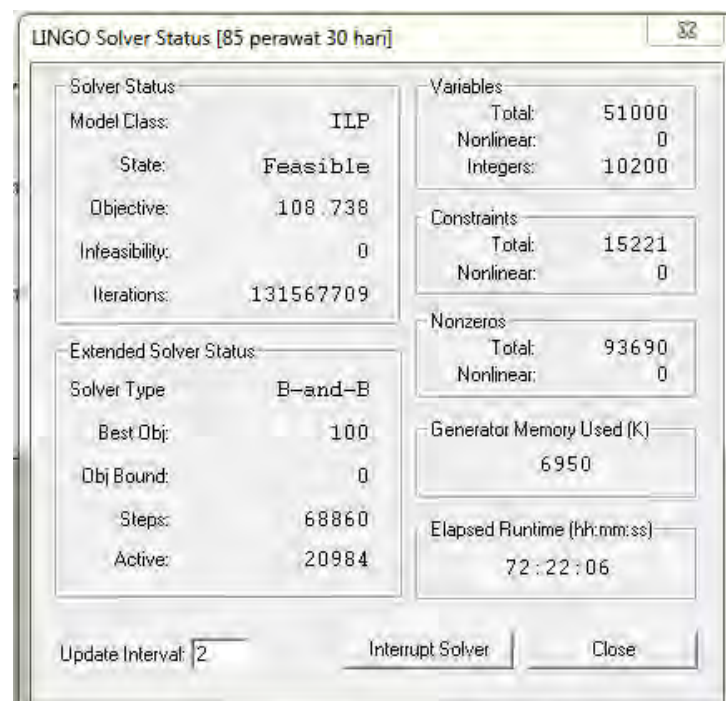
Replikasi Ke-	Waktu (detik)	Jumlah Workload Shift Pagi	Jumlah Workload Shift Sore	Jumlah Workload Shift Malam	Jumlah Workload Shift Libur	Pola Sore-Pagi	Pola Pagi-Malam	Pola Off-On-Off	Total
1	61,355	186	168	432	108	272	64	56	1286
2	61,021	168	168	408	84	268	76	80	1252
3	61,901	168	114	336	54	308	70	104	1154
4	61,355	150	138	390	126	280	90	44	1218
5	61,074	156	144	372	90	300	90	48	1200
6	61,433	192	99	384	117	264	60	56	1172

7	61,683	156	150	384	102	220	74	54	1140
8	61,753	156	108	324	72	368	98	96	1222
9	61,756	177	111	360	90	312	118	60	1228
10	61,544	189	156	396	75	316	82	26	1240

Dari tabel 5.14 didapatkan jika hasil eksperimen terbaik terdapat di replikasi ke-7 dengan total deviasi pelanggaran yang terjadi sebesar 1140. Jika dibandingkan dengan hasil eksperimen pada tabel 5.13, total deviasi pelanggaran yang terjadi masih lebih banyak dibandingkan dengan hasil eksperimen ini, maka hasil eksperimen ini yaitu dengan iterasi sebesar 50000 digunakan sebagai output solusi akhir dari eksperimen *simulated annealing*.

1.2.2 Eksperimen dengan Menggunakan Metode Eksak

Selain dilakukannya eksperimen dengan menggunakan algoritma *simulated annealing*, juga dilakukan eksperimen dengan menggunakan metode eksak dengan menggunakan *software* LINGO. Berikut ini merupakan hasil dari eksperimen dengan metode eksak dengan menggunakan *software* LINGO :



Gambar 5. 1 Status Eksperimen dengan *Software* LINGO

Karena keterbatasan waktu, maka eksperimen dari LINGO tersebut dihentikan setelah dilakukan selama 72 jam. Walaupun belum mencapai titik yang paling optimal tetapi status solusi yang didapat sudah *feasible*. Berikut ini merupakan rincian deviasi pelanggaran yang terjadi pada solusi akhir:

Tabel 5. 14 Rincian Deviasi Pelanggaran pada Metode Eksak (1)

Waktu (jam)	Jumlah Workload Shift Pagi	Jumlah Workload Shift Sore	Jumlah Workload Shift Malam	Jumlah Workload Shift Libur	Pola Sore-Pagi	Pola Pagi-Malam	Pola Off-On-Off	Total
72	6	3	3	3	56	18	186	275

Jumlah deviasi pelanggaran yang terjadi pada metode eksak berjumlah 275 dengan rinciannya adalah terdapat deviasi pelanggaran *workload* pada shift pagi sebesar 6, workload pada shift sore sebesar 3, workload pada shift malam sebesar 3, workload pada shift libur sebesar 3, pola shift sore diikuti shift pagi sebesar 56, pola shift pagi diikuti shift malam sebesar 18, dan pola “off-on-off” sebesar 186.

1.3 Perhitungan Kondisi Eksisting

Pada subbab ini dilakukan untuk menghitung berapa besar nilai deviasi pelanggaran yang terjadi pada kondisi eksisting, yaitu penjadwalan perawat pada RSUD Dr. Soetomo Surabaya. Berikut ini merupakan hasil dari perhitungan nilai deviasi pelanggaran yang terjadi pada kondisi eksisting:

Tabel 5. 15 Rincian Deviasi Pelanggaran Kondisi Eksisting (1)

Jeniss Pelanggaran	Jumlah Workload Shift Pagi	Jumlah Workload Shift Sore	Jumlah Workload Shift Malam	Jumlah Workload Shift Libur
Deviasi Pelanggaran	984	621	462	657
Bobot	3	3	3	3
Jumlah Pelanggaran	328	207	154	219

Tabel 5. 16 Rincian Deviasi Pelanggaran Kondisi Eksisting (2)

Jeniss Pelanggaran	Pola Setelah Shift Sore Diikuti Shift Pagi	Pola Setelah Shift Pagi Diikuti Shift Malam	Pola "Off-On-Off"	Total
Deviasi Pelanggaran	120	72	214	3130
Bobot	4	2	2	
Jumlah Pelanggaran	30	36	107	

Dari hasil pada tabel 5.17 dan tabel 5.18 didapatkan jika pada kondisi eksisting terdapat besar deviasi pelanggaran sebesar 3130 dimana rinciannya adalah terdapat deviasi pelanggaran *workload* pada shift pagi sebesar 984, workload pada shift sore sebesar 621, workload pada shift malam sebesar 462, workload pada shift libur sebesar 657, pola shift sore diikuti shift pagi sebesar 120, pola shift pagi diikuti shift malam sebesar 72, dan pola “off-on-off” sebesar 214. Dimana nantinya hasil kondisi eksisting dengan hasil eksperimen yang telah dilakukan akan dibandingkan solusi mana yang mempunyai nilai deviasi pelanggaran yang paling kecil dan digunakan sebagai dasar dalam perbaikan penjadwalan pada kondisi eksisting ini.

1.4 Perbandingan Hasil Pengujian

Setelah dilakukannya eksperimen dalam penyelesaian *nurse scheduling problem*, maka selanjutnya adalah dilakukannya perbandingan dari hasil pengujian yang sudah dilakukan. Perbandingan hasil pengujian yang dilakukan antara lain hasil pengujian dari *simulated annealing* dengan kondisi eksisting dan hasil *simulated annealing* dengan metode eksak.

1.4.1 Perbandingan Deviasi Pelanggaran Kondisi Eksisting dengan

Algoritma *Simulated Annealing*

Berikut ini merupakan hasil perbandingan deviasi pelanggaran kondisi eksisting dengan algoritma *simulated annealing* :

Tabel 5. 17 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting pada *Workload Shift Pagi*

Replikasi	<i>Simulated Annealing</i>	Kondisi	GAP (%)
-----------	----------------------------	---------	---------

Ke		Eksisting	
1	186	984	81,10
2	168	984	82,93
3	168	984	82,93
4	150	984	84,76
5	156	984	84,15
6	192	984	80,49
7	156	984	84,15
8	156	984	84,15
9	177	984	82,01
10	189	984	80,79

Dari tabel 5.18 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* pagi lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 82,74 %.

Tabel 5. 18 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Workload Shift sore

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	168	621	72,95
2	168	621	72,95
3	114	621	81,64
4	138	621	77,78
5	144	621	76,81
6	99	621	84,06
7	150	621	75,85
8	108	621	82,61
9	111	621	82,13
10	156	621	74,88

Dari tabel 5.19 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* sore lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 78,16 %.

Tabel 5. 19 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Workload Shift Malam

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	432	462	6,49

2	408	462	11,69
3	336	462	27,27
4	390	462	15,58
5	372	462	19,48
6	384	462	16,88
7	384	462	16,88
8	324	462	29,87
9	360	462	22,08
10	396	462	14,29

Dari tabel 5.20 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* malam lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 18,05 %.

Tabel 5. 20 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Workload Shift Libur

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	108	657	83,56
2	84	657	87,21
3	54	657	91,78
4	126	657	80,82
5	90	657	86,30
6	117	657	82,19
7	102	657	84,47
8	72	657	89,04
9	90	657	86,30
10	75	657	88,58

Dari tabel 5.21 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* libur lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 86,03 %.

Tabel 5. 21 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift Sore Diikuti Shift Pagi

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	272	120	126,67
2	268	120	123,33
3	308	120	156,67

4	280	120	133,33
5	300	120	150,00
6	264	120	120,00
7	220	120	83,33
8	368	120	206,67
9	312	120	160,00
10	316	120	163,33

Dari tabel 5.23 dapat dilihat jika hasil dari *simulated annealing* pada pola sore-pagi lebih buruk daripada kondisi eksistingnya dengan rata-rata GAP sebesar 142,33%.

Tabel 5. 22 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift Pagi Diikuti Shift Malam

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	64	72	11,11
2	76	72	5,56
3	70	72	2,78
4	90	72	25,00
5	90	72	25,00
6	60	72	16,67
7	74	72	2,78
8	98	72	36,11
9	118	72	63,89
10	82	72	13,89

Dari tabel 5.24 dapat dilihat jika hasil dari *simulated annealing* pada pola pagi-malam lebih buruk daripada kondisi eksistingnya dengan rata-rata GAP sebesar 20,28 %.

Tabel 5. 23 Perbandingan Jumlah Deviasi Pelanggaran SA dan Kondisi Eksisting Pola Shift “Off-On-Off”

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	56	214	73,83
2	80	214	62,62
3	104	214	51,40
4	44	214	79,44
5	48	214	77,57
6	56	214	73,83
7	54	214	74,77

8	96	214	55,14
9	60	214	71,96
10	26	214	87,85

Dari tabel 5.25 dapat dilihat jika hasil dari *simulated annealing* pada pola *shift* “off-on-off” lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 70,84 %.

Tabel 5. 24 Perbandingan Total Deviasi Pelanggaran yang Terjadi

Replikasi Ke	<i>Simulated Annealing</i>	Kondisi Eksisting	GAP (%)
1	1286	3130	58,91
2	1252	3130	60,00
3	1154	3130	63,13
4	1218	3130	61,09
5	1200	3130	61,66
6	1172	3130	62,56
7	1140	3130	63,58
8	1222	3130	60,96
9	1228	3130	60,77
10	1240	3130	60,38

Dari tabel 5.26 dapat dilihat jika hasil dari *simulated annealing* secara keseluruhan lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 61,30 %.

1.4.2 Perbandingan Deviasi Pelanggaran dengan Metode Eksak dan Kondisi Eksisting

Berikut ini merupakan hasil perbandingan deviasi pelanggaran metode eksak dengan kondisi eksisting :

Tabel 5. 25 Perbandingan Metode Eksak dan Kondisi Eksisting

Jenis Pelanggaran	Metode Eksak	Kondisi Eksisting	GAP (%)
Jumlah Workload Shift Pagi	6	984	99,39
Jumlah Workload Shift Sore	3	621	99,52
Jumlah Workload Shift Malam	3	462	99,35

Jumlah Workload Shift Libur	3	657	99,54
Pola Sore-Pagi	56	120	53,33

Tabel 5. 26 Perbandingan Metode Eksak dan Kondisi Eksisting (cont)

Jenis Pelanggaran	Metode Eksak	Kondisi Eksisting	GAP (%)
Pola Pagi-Malam	18	72	75
Pola Off-On-Off	186	214	13,08
Total	275	3130	91,21

Dari hasil yang didapat dengan menggunakan metode eksak pada tabel 5.27 ini dapat dilihat satu persatu jenis pelanggaran yang jika hasil yang didapat jauh lebih baik daripada kondisi eksistingnya, yaitu dapat memperbaiki kondisi eksistingnya sebesar 91,21 %

1.4.3 Perbandingan Deviasi Pelanggaran dengan Algoritma *Simulated*

Annealing dan Metode Eksak

Berikut ini merupakan hasil perbandingan deviasi pelanggaran menggunakan algoritma *simulated annealing* dan metode eksak:

Tabel 5. 27 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak *Workload Shift Pagi*

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	186	6	3000,00
2	168	6	2700,00
3	168	6	2700,00
4	150	6	2400,00
5	156	6	2500,00
6	192	6	3100,00
7	156	6	2500,00
8	156	6	2500,00
9	177	6	2850,00
10	189	6	3050,00

Dari tabel 5.28 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* pagi lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 2730 %.

Tabel 5. 28 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift sore

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	168	3	5500,00
2	168	3	5500,00
3	114	3	3700,00
4	138	3	4500,00
5	144	3	4700,00
6	99	3	3200,00
7	150	3	4900,00
8	108	3	3500,00
9	111	3	3600,00
10	156	3	5100,00

Dari tabel 5.29 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* sore lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 4420 %.

Tabel 5. 29 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift Malam

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	432	3	14300,00
2	408	3	13500,00
3	336	3	11100,00
4	390	3	12900,00
5	372	3	12300,00
6	384	3	12700,00
7	384	3	12700,00
8	324	3	10700,00
9	360	3	11900,00
10	396	3	13100,00

Dari tabel 5.30 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* malam lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 12520 %.

Tabel 5. 30 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Workload Shift Libur

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	108	56	92,86
2	84	56	50,00
3	54	56	3,57
4	126	56	125,00
5	90	56	60,71
6	117	56	108,93
7	102	56	82,14
8	72	56	28,57
9	90	56	60,71
10	75	56	33,93

Dari tabel 5.31 dapat dilihat jika hasil dari *simulated annealing* pada *workload shift* libur lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 64,64 %.

Tabel 5. 31 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift Sore Diikuti Shift Pagi

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	272	18	1411,11
2	268	18	1388,89
3	308	18	1611,11
4	280	18	1455,56
5	300	18	1566,67
6	264	18	1366,67
7	220	18	1122,22
8	368	18	1944,44
9	312	18	1633,33
10	316	18	1655,56

Dari tabel 5.32 dapat dilihat jika hasil dari *simulated annealing* pada pola *shift* sore-pagi lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 1515,56%.

Tabel 5. 32 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift Pagi Diikuti Shift Malam

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	64	18	255,56
2	76	18	322,22
3	70	18	288,89
4	90	18	400,00
5	90	18	400,00
6	60	18	233,33
7	74	18	311,11
8	98	18	444,44
9	118	18	555,56
10	82	18	355,56

Dari tabel 5.33 dapat dilihat jika hasil dari *simulated annealing* pada pola *shift* pagi-malam lebih buruk daripada metode eksak dengan rata-rata GAP sebesar 356,67%.

Tabel 5. 33 Perbandingan Jumlah Deviasi Pelanggaran SA dan Metode Eksak Pola Shift “Off-On-Off”

Replikasi Ke	<i>Simulated Annealing</i>	Metode Eksak	GAP (%)
1	56	186	69,89
2	80	186	56,99
3	104	186	44,09
4	44	186	76,34
5	48	186	74,19
6	56	186	69,89
7	54	186	70,97
8	96	186	48,39
9	60	186	67,74
10	26	186	86,02

Dari tabel 5.34 dapat dilihat jika hasil dari *simulated annealing* pada pola *shift* “off-on-off” lebih baik daripada kondisi eksistingnya dengan rata-rata GAP sebesar 66,45 %.

Tabel 5. 34 Perbandingan SA dan Metode Eksak pada Waktu Komputasi

Replikasi Ke	<i>Simulated Annealing (jam)</i>	Metode Eksak (jam)	GAP (%)
--------------	----------------------------------	--------------------	---------

1	0,017043111	72	99,9763
2	0,016950278	72	99,9765
3	0,017194778	72	99,9761
4	0,017043111	72	99,9763
5	0,016965111	72	99,9764
6	0,017064778	72	99,9763
7	0,017134111	72	99,9762
8	0,017153472	72	99,9762
9	0,017154361	72	99,9762
10	0,017095472	72	99,9763

Dari tabel 5.35 dapat dilihat jika dengan menggunakan *simulated annealing* maka hanya membutuhkan waktu rata-rata 0,017 jam atau 61 detik untuk menyelesaikan *nurse scheduling problem*. Bandingkan dengan metode eksak yang membutuhkan waktu 72 jam dalam penyelesaiannya.

1.5 Analisis Hasil Eksperimen

Dalam bagian ini akan dilakukan analisis terkait hasil dari beberapa eksperimen yang telah dilakukan dalam menyelesaikan *nurse scheduling problem*.

1.5.1 Analisis Hasil Eksperimen dengan Algoritma *Simulated Annealing*

Pada subbab ini akan dilakukan analisis terkait hasil eksperimen yang telah dilakukan dengan algoritma *simulated annealing* pada *nurse scheduling problem*

1.5.1.1 Analisis Uji Parameter pada Algoritma *Simulated Annealing*

Sebelum melakukan eksperimen dengan menggunakan algoritma *simulated annealing*, maka langkah yang harus dilakukan terlebih dahulu adalah dengan menentukan nilai parameter awal yang akan digunakan dalam eksperimen yang gunanya adalah untuk mendapatkan solusi yang paling baik dengan waktu komputasi yang cepat. Dalam melakukan uji parameter, parameter-parameter yang diuji adalah faktor reduksi temperatur (Cr), jumlah siklus penurunan temperatur (N), dan temperatur awal (To).

Uji parameter pertama yang dilakukan adalah dengan menguji faktor reduksi temperatur (Cr). Pada saat melakukan uji parameter faktor reduksi temperatur, skenario perubahan yang digunakan adalah 0,5, 0,7, 0,9 dan 0,95

dengan temperatur awal dan jumlah siklus penurunan temperatur yang tetap, yaitu 500 dan 5. Dari eksperimen uji parameter yang dilakukan didapatkan jika dengan menggunakan Cr sebesar 0,95 maka mendapatkan rata-rata jumlah deviasi pelanggaran yang paling minimal. Hal ini dikarenakan jika dibandingkan dengan Cr yang lebih kecil dari 0,95, maka dalam melakukan penurunan temperatur akan semakin cepat. Karena penurunan temperatur yang cepat maka iterasi yang dibutuhkan dalam mendapatkan solusi akhir juga semakin sedikit yang berakibat semakin sempitnya ruang pencarian solusi sehingga menyebabkan pada solusi akhir yang buruk. Walaupun waktu komputasinya menjadi lebih lamban tetapi itu merupakan sebuah *trade-off* dimana jika menginginkan solusi yang lebih baik, maka membutuhkan waktu yang lebih lama dalam melakukan komputasi.

Uji parameter kedua yang dilakukan adalah dengan menguji jumlah siklus penurunan temperatur (N) dengan menggunakan skenario N sebesar 5,10,15 dan dengan temperatur tetap yaitu 500 dan Cr yang terbaik pada pengujian sebelumnya yaitu 0,95. Hal yang sama juga terjadi pada uji parameter kedua ini dimana dengan N sebesar 15 mendapatkan solusi akhir yang lebih baik daripada menggunakan N sebesar 5 dan 10. Tetapi waktu komputasi yang digunakan menjadi lebih lamban dikarenakan penurunan temperatur dilakukan setiap 15 iterasi. Sehingga ruang dalam pencapaian solusi akhir menjadi sangat besar yang menyebabkan konvergensi dari solusi ini menjadi lebih lama dan peluang solusi terjebak ke dalam lokal optimal menjadi sangat minim sehingga mendapatkan solusi akhir yang lebih baik jika dibandingkan dengan jumlah N yang lain.

Lalu uji parameter yang ketiga adalah dengan menguji temperatur awal (T_0) yang digunakan dengan skenario T_0 sebesar 2000,5000 dan 10000 dengan Cr sebesar 0,95 dan N sebesar 15. Hasil uji parameter T_0 didapatkan jika dengan menggunakan T_0 sebesar 10000 mendapatkan solusi akhir yang paling bagus. Walaupun hasil yang didapat dari ketiga skenario ini sebenarnya tidak jauh berbeda hal ini dikarenakan beda temperatur yang digunakan tiap skenario tidak terlalu jauh sehingga jumlah iterasi yang dibutuhkan untuk mendapatkan solusi akhir tidak terlalu jauh juga tetapi dikarenakan masih adanya perbedaan iterasi yang dibutuhkan dalam pencarian solusi dan menyebabkan ruang dalam

mendapatkan solusi akhir menjadi lebih besar sehingga dengan menggunakan To sebesar 10000 mendapatkan solusi akhir yang paling baik.

1.5.1.2 Analisis Hasil Eksperimen dengan Algoritma *Simulated Annealing*

Setelah didapatkan parameter awal setelah dilakukannya uji parameter maka dilakukan eksperimen pertama algoritma *simulated annealing* pada *nurse scheduling problem* dengan *stopping criteria* adalah minimal temperatur sebesar 10^{-6} . Setelah dilakukan eksperimen sebanyak 10 kali maka didapatkan jika jumlah deviasi pelanggaran yang terjadi terdapat pada kisaran antara 1652 dan 1762 dimana solusi akhir yang terbaik terjadi pada replikasi ke 10 yaitu dengan jumlah deviasi pelanggaran sebesar 1652. Lalu untuk membandingkan hasil eksperimen tersebut, dilakukan eksperimen kedua dengan mengganti *stopping criteria* menjadi jumlah iterasi sebesar 50.000 dimana hasil dari eksperimen kedua ini jauh lebih baik daripada eksperimen pertama yaitu jumlah deviasi pelanggaran paling kecil sebesar 1140. Hal ini dikarenakan semakin besarnya jumlah iterasi, maka akan membantu mencapai keseimbangan termal pada setiap temperatur tetapi akan memakan waktu komputasi yang lama. Dengan besarnya jumlah iterasi maka konvergensi yang terjadi akan semakin lama sehingga proses pencarian solusi menjadi semakin baik dikarenakan tidak ada titik-titik potensial yang bisa menjadi global optimal yang terlewat.

1.5.2 Analisis Eksperimen dengan Menggunakan Metode Eksak

Dalam menggunakan metode eksak, metode yang digunakan dalam eksperimen adalah metode *branch and bound*. Hasil dari eksperimen ini sebenarnya jika dilihat dari status LINGO belum mencapai titik optimal hal ini dikarenakan masih harus menunggu waktu yang lebih lama lagi untuk mencapai titik optimal tersebut. Dari eksperimen yang dilakukan, waktu komputasi sudah lebih dari 72 jam dan hasil yang didapat pun masih merupakan *feasible solution*. Dan itu merupakan kelemahan dari metode eksak dimana dalam melakukan penyelesaian permasalahan yang besar akan membutuhkan waktu yang sangat lama untuk mendapatkan solusi yang optimal. Walaupun solusi yang akan didapat merupakan solusi yang paling optimal, tetapi membutuhkan waktu komputasi yang lama untuk mendapatkan solusi tersebut.

1.5.3 Analisis Perbandingan Deviasi Pelanggaran dengan *Simulated Annealing* dan Kondisi Eksisting

Jika dilihat total deviasi pelanggaran dari algoritma *simulated annealing* dibandingkan dengan kondisi eksisting maka dapat disimpulkan jika dengan menggunakan algoritma *simulated annealing* dalam penyelesaian *nurse scheduling problem*, maka bisa mendapatkan hasil yang jauh yang lebih baik daripada kondisi eksistingnya. Sehingga bisa dilakukan perbaikan penjadwalan antara penjadwalan kondisi eksistingnya dengan hasil yang didapat dari *simulated annealing*.

Performansi dari *simulated annealing* jika dilihat satu persatu dimulai dari workload pada shift pagi, shift sore, shift malam, dan shift libur dimana GAP yang terjadi jauh lebih baik daripada kondisi eksisting yaitu dengan rata-rata sebesar 82,74 % pada shift pagi, 78,16 % pada shift sore, 18,05 % pada shift malam, dan 8,03 % pada shift libur. Jika dilihat dari semakin baiknya hasil yang didapat dibandingkan kondisi eksisting hal ini menandakan jika jumlah workload yang diterima tiap perawat di rumah sakit menjadi *balance* sehingga sangat sedikit terjadinya ketimpangan jumlah workload antara perawat satu dengan perawat lain. Lalu jika dilihat dari *constraint* lain yaitu pada pola “off-on-off” terlihat hasil *simulated annealing* menunjukkan hasil yang lebih baik dibandingkan dengan kondisi eksistingnya

Tetapi keadaan yang berbeda jika dilihat dari pola shift sore-pagi dan pola shift pagi-malam dimana terjadi perbedaan yang antara hasil dari *simulated annealing* dengan kondisi eksisting. Dari hasil tersebut didapatkan jika GAP hasil *simulated annealing* tidak lebih baik daripada kondisi eksisting. Hal ini dikarenakan dalam melakukan penyelesaian dengan menggunakan *simulated annealing* belum mencapai titik yang paling optimal, tetapi hanya mendekati optimal sehingga hasil yang didapat terkadang masih sangat buruk. Seperti halnya hasil yang didapat pada permasalahan ini dimana hasil dari *simulated annealing* tidak dapat memperbaiki kondisi eksistingnya. Tetapi jika dilihat dari total deviasi pelanggaran yang terjadi maka bisa menunjukkan dengan menggunakan *simulated annealing* dapat memperbaiki penjadwalan pada kondisi eksisting sebesar 61,30 % sehingga hanya dengan waktu rata-rata 61,48 detik dalam penyelesaian

masalah, maka performansi solusi akhir dari *simulated annealing* sudah cukup untuk dapat memperbaiki kondisi eksistingnya.

1.5.4 Analisis Perbandingan Deviasi Pelanggaran dengan Metode Eksak dan Kondisi Eksisting

Jika dilihat dari hasil perbandingan jumlah deviasi pelanggaran yang terjadi, maka dapat dilihat jika hasil dari metode eksak dapat memperbaiki secara signifikan dari kondisi eksistingnya. Bisa dilihat jika besar perbaikan total yang dilakukan mencapai lebih dari 90 % dimana hanya sedikit pelanggaran yang terjadi pada hasil metode eksak ini. Walaupun hasil dari metode eksak dapat memperbaiki secara signifikan pada kondisi eksistingnya, tetapi waktu komputasi yang harus dilewati sangat lama. Hal ini dikarenakan dengan metode eksak akan selalu mencari solusi yang global optimum sehingga akan membutuhkan waktu yang sangat lama untuk mencoba semua kemungkinan solusi yang didapatkan terlebih jika kasus tersebut merupakan kasus yang berskala besar. Oleh karena itu jika dilihat dari segi solusi yang didapat maka performansi dari metode eksak sangat baik, tetapi jika dilihat dari segi waktu komputasi dalam penyelesaiannya, maka performansi dari metode eksak ini sangatlah buruk.

1.5.5 Analisis Perbandingan Deviasi Pelanggaran dengan Algoritma *Simulated Annealing* dan Menggunakan Metode Eksak

Dari hasil penyelesaian *nurse scheduling problem* menggunakan metode eksak dan *simulated annealing* didapatkan jika dengan menggunakan metode eksak jumlah deviasi pelanggaran yang terjadi jauh lebih sedikit daripada *simulated annealing*.

Hasil eksperimen menunjukkan jika terjadi GAP yang sangat besar pada *workload* perawat pada shift pagi, shift sore, shift malam, shift libur, pola shift sore-pagi, dan pola shift pagi-malam. Dari hasil tersebut menunjukkan jika hasil *simulated annealing* lebih buruk daripada hasil dari metode eksak. walaupun jika dilihat pada pola shift “off-on-off” menunjukkan jika *simulated annealing* lebih baik tetapi secara keseluruhan solusi akhir dari *simulated annealing* lebih buruk jika dibandingkan dengan metode eksak

Hal ini merupakan sangat wajar dikarenakan solusi akhir *simulated annealing* ini hanya mendekati optimal sehingga membutuhkan waktu komputasi yang tidak lama. Bandingkan jika dilakukan dengan menggunakan metode eksak dimana waktu komputasi dalam eksperimen sudah lebih dari 72 jam dan itu masih belum mencapai titik optimal hanya mencapai solusi yang *feasible* saja. Hal itu dikarenakan dalam metode eksak dalam menggenerate solusi jauh lebih lambat daripada *simulated annealing*, terlebih kasus yang digunakan dalam *nurse scheduling problem* merupakan kasus dengan skala yang besar sehingga akan membutuhkan waktu yang sangat lama dalam mencapai titik yang optimal. Oleh karena itu perfomansi dari *simulated annealing* masih sangat baik jika dibutuhkan waktu penyelesaian yang sangat singkat.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan hasil penelitian dan saran-saran yang berkaitan dengan penelitian selanjutnya

1.1 Kesimpulan

Berdasarkan hasil eksperimen dan analisis yang telah dilakukan pada bab sebelumnya maka kesimpulan yang dapat diambil dari penelitian ini adalah :

1. Dalam penelitian ini dilakukan penyelesaian *nurse scheduling problem* dengan metode eksak dan metaheuristik. Hasil yang didapat sudah dapat dilakukan perbaikan terhadap kondisi eksisting. Tetapi kedua metode tersebut mempunyai kelebihan dan kekurangan dalam penyelesaian NSP yaitu jika menggunakan metode eksak hasil yang dapat merupakan hasil yang global optimal tetapi membutuhkan waktu yang sangat lama dalam mendapatkan solusi. Jika menggunakan metaheuristik hasil yang didapat hanya mendekati optimal tetapi hanya membutuhkan waktu yang tidak lama dalam mendapatkan solusi tersebut.
2. Dari penelitian ini didapatkan hasil NSP dengan jumlah deviasi pelanggaran yang terjadi sebesar 1140 pelanggaran dengan menggunakan metaheuristik dan sebesar 275 dengan menggunakan metode eksak. Jika dibandingkan dengan kondisi eksisting yang terdapat pada rumah sakit yang memiliki pelanggaran sebesar 3130, maka dengan hasil NSP dari kedua metode tersebut sudah dapat dijadikan solusi perbaikan dalam penjadwalan pada rumah sakit tersebut.
3. Kombinasi parameter yang paling baik untuk mendapatkan hasil *nurse scheduling problem* yang optimal adalah dengan menggunakan faktor pereduksi temperatur (Cr) sebesar 0,95, jumlah siklus penurunan temperatur sebesar 15, dan temperatur awal sebesar 15000.

1.2 Saran

Adapun saran yang dapat diberikan dalam penelitian ini untuk penelitian selanjutnya adalah :

1. Menggunakan algoritma metaheuristik lain untuk membandingkan performansi algoritma yang terbaik yang bisa digunakan dalam penyelesaian *nurse scheduling problem*.
2. Untuk pengembangan permasalahan ini, mempertimbangkan level dari perawat dan tenaga kerja lain seperti pembantu perawat dan petugas laboratorium.

DAFTAR PUSTAKA

A.T. Ernst, h. j. 2004. *staff scheduling and rostering: a review of applications, methods and models. european journal of operational research*, 4-27.

Asmadi 2008. *konsep dasar keperawatan*. Jakarta : buku kedokteran egc

Husen, A. 2008. *manajemen proyek*. Yogyakarta : c.v andi offset

Ko,Y.W. 2013. *an improvemen technique for simulated annealing and its application to nurse scheduling problem. iinternational journal*, 7.

Morton, T. E. & Pentico, D. W. 1993. *heuristic scheduling systems: withapplication in manufacturing and services*, New York.

Santosa, B & Willy, P. 2011. *metoda metaheuristik : konsep dan implementasi*. surabaya: guna widya

M'hallah, Rym & Alkhabbaz, Amina. 2013. *scheduling of nurses : a case study of a kuwaiti health care unit. 1-19*

Blochlinger, Ivo. 2003. *modelling staff scheduling. a tutorial. european journal of operational research. european journal of operational research* 533-542

Jenal, Ruzakkiah et al. 2011. *a cyclical nurse schedule using goal programming* 151-164

Ismail, W.R. et al. 2012. *goal programming based master plan for cyclical nurse scheduling. journal of theoritical and applied information technology. i.*

Kundu, S et al. 2008. *comparative performance of simulated annealing and genetic algorithm in solving nurse scheduling problem. proceedings of the international multi conference of engineers and computer scientist. 19-21*

LAMPIRAN A

Pada lampiran ini akan diberikan coding algoritma *simulated annealing* dan model matematis kedalam bahasa MATLAB dan LINGO

Coding LINGO

```
sets:

perawat/1..85/;

hari/1..30/;

aaa(hari,perawat):xp,xq,xr,xo,u1,u2,u3,u4,u5,u6,u7,e1,e2,e3,e4,e5,e6,e7,e8,u8;

endsets

data:

@ole('E:\lingoku\rekap.xlsx')=xp,xq,xr,xo;

enddata

min= (3*@sum(aaa(i,j)|i#EQ#j:u1(i,j)))
+ (3*@sum(aaa(i,j)|i#EQ#j:u2(i,j)))+(3*@sum(aaa(i,j)|i#EQ#j:u3(i,j)))
+ (3*@sum(aaa(i,j)|i#EQ#j:e1(i,j)))+(3*@sum(aaa(i,j)|i#EQ#j:e2(i,j)))
+ (3*@sum(aaa(i,j)|i#EQ#j:e3(i,j)))+(3*@sum(aaa(i,j):u4(i,j)))+(3*@sum(aaa(i,j):e4(i,j)))+(2*@sum(aaa(i,j):e5(i,j)))+(4*@sum(aaa(i,j):e6(i,j)))+(2*@sum(aaa(i,j):e7(i,j)));

!jumlah perawat minimal dan maksimal per shift;

@for(hari(i):@sum(aaa(i,j):xp(i,j))>=25);

@for(hari(i):@sum(aaa(i,j):xq(i,j))>=17);

@for(hari(i):@sum(aaa(i,j):xr(i,j))>=15);

!hanya ada 1 shift perhari per perawat;

@for(perawat(j):@for(hari(i):@sum(aaa(i,j):xp(i,j)+xq(i,j)+xr(i,j)+xo(i,j))=1));
```

```

!consecutive malam;

@for(perawat(j):@for(hari(i)|i#EQ#1#or#i#EQ#3#or#i#EQ#5#or#i#EQ#7#or
#i#EQ#9#or#i#EQ#11#or#i#EQ#13#or#i#EQ#15#or#i#EQ#17#or#i#EQ#19#or#i#
EQ#21#or#i#EQ#23#or#i#EQ#25#or#i#EQ#27

:xr(i,j)<=xr(i+1,j)));

@for(perawat(j):@for(hari(i)|i#EQ#1#or#i#EQ#3#or#i#EQ#5#or#i#EQ#7#or
#i#EQ#9#or#i#EQ#11#or#i#EQ#13#or#i#EQ#15#or#i#EQ#17#or#i#EQ#19#or#i#
EQ#21#or#i#EQ#23#or#i#EQ#25#or#i#EQ#27

:1-xr(i,j)<=10000*(1-xr(i+1,j))));

@for(perawat(j):@for(hari(i)|i#EQ#1#or#i#EQ#3#or#i#EQ#5#or#i#EQ#7#or
#i#EQ#9#or#i#EQ#11#or#i#EQ#13#or#i#EQ#15#or#i#EQ#17#or#i#EQ#19#or#i#
EQ#21#or#i#EQ#23#or#i#EQ#25#or#i#EQ#27

:xr(i+1,j)<=xo(i+2,j)));

@for(perawat(j):@for(hari(i)|i#EQ#1#or#i#EQ#3#or#i#EQ#5#or#i#EQ#7#or
#i#EQ#9#or#i#EQ#11#or#i#EQ#13#or#i#EQ#15#or#i#EQ#17#or#i#EQ#19#or#i#
EQ#21#or#i#EQ#23#or#i#EQ#25#or#i#EQ#27

:xr(i,j)+xr(i+1,j)=2*xo(i+2,j)));

@for(perawat(j):@for(hari(i)|i#EQ#29:xr(i,j)<=xr(i+1,j)));

@for(perawat(j):@for(hari(i)|i#EQ#29:1-xr(i,j)<=10000*(1-
xr(i+1,j))));

!workload perawat;

@for(perawat(j):@sum(aaa(i,j):xp(i,j))+
@sum(aaa(i,j)|i#EQ#j:u1(i,j))-@sum(aaa(i,j)|i#EQ#j:e1(i,j))=9);

@for(perawat(j):@sum(aaa(i,j):xq(i,j))+
@sum(aaa(i,j)|i#EQ#j:u2(i,j))-@sum(aaa(i,j)|i#EQ#j:e2(i,j))=8);

@for(perawat(j):@sum(aaa(i,j):xr(i,j))+
@sum(aaa(i,j)|i#EQ#j:u3(i,j))-@sum(aaa(i,j)|i#EQ#j:e3(i,j))=6);

!jumlah libur perawat;

```

```

@for(perawat(j):@sum(aaa(i,j):xo(i,j))+ @sum(aaa(i,j):u4(i,j))-
@sum(aaa(i,j):e4(i,j))=7);

!menghindari shift kerja off-on-off;

@for(perawat(j):@for(hari(i)|i#LE#28:@sum(aaa(i,j):xo(i,j)+xp(i+1,j)
+xq(i+1,j)+xr(i+1,j)+xo(i+2,j))+ @sum(aaa(i,j):u5(i,j))-
@sum(aaa(i,j):e5(i,j))=2));

!menghindari setelah shift sore diikuti shift pagi;

@for(perawat(j):@for(hari(i)|i#LE#29:@sum(aaa(i,j):xq(i,j)+xp(i+1,j)
) + @sum(aaa(i,j):u6(i,j))-@sum(aaa(i,j):e6(i,j))=1));

!menghindari setelah shift pagi diikuti shift malam;

@for(perawat(j):@for(hari(i)|i#LE#29:@sum(aaa(i,j):xp(i,j)+xr(i+1,j)
) + @sum(aaa(i,j):u7(i,j))-@sum(aaa(i,j):e7(i,j))=1));

@for(aaa(i,j):@bin(xp(i,j)));

@for(aaa(i,j):@bin(xq(i,j)));

@for(aaa(i,j):@bin(xr(i,j)));

@for(aaa(i,j):@bin(xo(i,j)));

```

Coding Matlab

```
function
[h,z,pl_pagi,pl_sore,pl_malam,pl_libur,pl_sorepagi,pl_pagimalam,pl_o
ffonoff,iterasi,waktukomputasi]=sanspex4(perawat,hari,wp,ws,wm,wl,te
mpawal,cr,maxit)

waktukomputasi=cputime;
temperatur=tempawal;
siklus=0;
iterasi=0;

h=zeros(perawat,hari);
it=0;
i=1;
k=1;
for j=1:hari
    i=1;
while k<=hari
    cek=1;
    it=0;
while cek==1
    it=it+1;
    if k==1
        m=floor(rand*perawat);
        jum_malam(1,k)=m;
        p=floor(rand*perawat-m);
        jum_pagi(1,k)=p;
        s=floor(rand*(perawat-p-m));
        jum_sore(1,k)=s;
        jum_libur(1,k)=perawat-p-s-m;
        rekap_pagi(1,k)=jum_pagi(1,k);
        rekap_sore(1,k)=jum_sore(1,k);
        rekap_malam(1,k)=jum_malam(1,k);
        rekap_libur(1,k)=jum_libur(1,k);

    elseif mod(k,2)==0
        jum_malam(1,k)=jum_malam(1,k-1);
        m=jum_malam(1,k);
        p=floor(rand*(perawat-m));
        jum_pagi(1,k)=p;
        s=floor(rand*(perawat-m-p));
        jum_sore(1,k)=s;
        jum_libur(1,k)=perawat-m-p-s;
        rekap_pagi(1,k)=jum_pagi(1,k);
        rekap_sore(1,k)=jum_sore(1,k);
        rekap_malam(1,k)=jum_malam(1,k);
        rekap_libur(1,k)=jum_libur(1,k);
    else
        jum_libur(1,k)=jum_malam(1,k-1);
        l=jum_libur(1,k);
```

```

        m=floor(rand*perawat);
        jum_malam(1,k)=m;
        p=floor(rand*perawat-l-m);
        jum_pagi(1,k)=p;
        s=floor(rand*(perawat-p-l-m));
        jum_sore(1,k)=s;
        jum_libur(1,k)=perawat-p-s-m;
        rekap_pagi(1,k)=jum_pagi(1,k);
        rekap_sore(1,k)=jum_sore(1,k);
        rekap_malam(1,k)=jum_malam(1,k);
        rekap_libur(1,k)=jum_libur(1,k);
    end

    if jum_pagi(1,k)<25 || jum_sore(1,k)<17 || jum_malam(1,k)<15
    || jum_malam(1,k)>20 || jum_sore(1,k)>22 || jum_pagi(1,k)>29
        cek=1;
    else
        cek=0;
        k=k+1;
    end
    it=0;
end
end
end
jum_pagi=rekap_pagi;
jum_sore=rekap_sore;
jum_malam=rekap_malam;
jum_libur=rekap_libur;

%%solusi malam
j=1;
byk=1;
while byk<=15
    f=jum_malam(j);
    i=1;
    while i<=f
        ke=ceil(rand*perawat);
        benar=1;
        if byk==1
            while benar==1
                if h(ke,j)==3
                    benar=1;
                    ke=ceil(rand*perawat);
                    it=it+1;
                else
                    benar=0;
                    i=i+1;
                end
            end
            h(ke,j)=3;
            h(ke,j+1)=3;
            h(ke,j+2)=4;
        end
    end
end

```

```

if byk<=5 && byk~=1
while benar==1
    mlm=length(find(h(ke,:)==3));
    if mlm>4 && mlm<=6
        benar=1;
        ke=ceil(rand*perawat);
        it=it+1;

        elseif h(ke,j)==4
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        elseif h(ke,j)==3
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        else
            benar=0;
            i=i+1;
        end
    end
    h(ke,j)=3;
    h(ke,j+1)=3;
    h(ke,j+2)=4;
end
if byk>5 && byk<=10
while benar==1
    mlm=length(find(h(ke,:)==3));
    if mlm>4 && mlm<=6
        benar=1;
        ke=ceil(rand*perawat);
        it=it+1;

        elseif h(ke,j)==4
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        elseif h(ke,j)==3
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        else
            benar=0;
            i=i+1;
        end
    end
    h(ke,j)=3;
    h(ke,j+1)=3;
    h(ke,j+2)=4;
end
if byk>10 && byk<=15
while benar==1
    mlm=length(find(h(ke,:)==3));

```

```

        if mlm>6 && mlm<=8
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;

        elseif h(ke,j)==4
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        elseif h(ke,j)==3
            benar=1;
            ke=ceil(rand*perawat);
            it=it+1;
        else
            benar=0;
            i=i+1;
        end
    end
    h(ke,j)=3;
    h(ke,j+1)=3;
    h(ke,j+2)=4;
end
end

j=j+2;
byk=byk+1;

end
h=h(:,1:30);
%%% menghapus jumlah libur yang sudah digunakan
for j=1:hari
    i=1;
    if j~=1 && mod(j,2)==1
        bantu=find(h(:,j-1)==3);
        for m=1:length(bantu)
            h(bantu(m),j)=4;
        end
        jum_libur(1,j)=jum_libur(1,j)-length(bantu);
    end
end

%solusi pagi sore libur + malam
i=1;
for j=1:hari
    i=1;
    while i<=perawat
        r=rand;

        if r<0.333
            if h(i,j)==3
                i=i+1;
            elseif h(i,j)==0 && jum_pagi(1,j)>=1

```



```

        h(i,j)=1;
        jum_pagi(1,j)=jum_pagi(1,j)-1;
        i=i+1;
    elseif h(i,j)==4
        i=i+1;
    end

end

if r>0.333 && r<0.667
    if h(i,j)==3
        i=i+1;
    elseif h(i,j)==0 && jum_sore(1,j)>=1
        h(i,j)=2;
        jum_sore(1,j)=jum_sore(1,j)-1;
        i=i+1;
    elseif h(i,j)==4
        i=i+1;
    end
end
if r>0.667 && r<1
    if h(i,j)==3
        i=i+1;
    elseif h(i,j)==0 && jum_libur(1,j)>=1
        h(i,j)=4;
        jum_libur(1,j)=jum_libur(1,j)-1;
        i=i+1;
    elseif h(i,j)==4
        i=i+1;
    end
end
end
end

hawal=h;
[zawal,zrawal]=obj(hawal,wp,ws,wm,wl,hari,perawat);
zlama=zawal;
zrlama=zrawal;
hlama=hawal;

%%fungsi simulated annealing
grafik=0;
while iterasi < maxit
    %%mencari solusi baru
    temph=hlama;
    hbaru=hlama;
    ke=ceil(hari*rand);
    ins=sort(ceil(perawat*rand(1,2)));
    s=ins(1);
    t=ins(2);

    r=rand;

```

```

%%mencari solusi baru dengan slide swap flip
if r <.333
    %flip
    hbaru=hbaru';
    temp=hbaru;
    hbaru(ke,s:t)=fliplr(temp(ke,s:t));
    hbaru=hbaru';
    temp=temp';
elseif r <.667 && r > 0.333
    %swap
    hbaru([s t],ke) = temp([t s],ke);
else
    %slide
    hbaru(s:t,ke) = temp([s+1:t s],ke);
end

hbaru=ceksolusi(hbaru,ke);
[zbaru,zrbaru]=obj(hbaru,wp,ws,wm,wl,hari,perawat);
%membandingkan z baru dan z lama
diff=abs(zbaru-zlama);

%menurunkan temperatur jika sudah mencapai siklus
if siklus>=15
    temperatur=cr*temperatur;
    siklus=0;
end

%z baru lebih baik daripada z lama
if zbaru<zlama
    hlama=hbaru;
    zlama=zbaru;
    zrlama=zrbaru;
    siklus=siklus+1;
    iterasi=iterasi+1;
    %z baru tidak lebih baik dari z lama
else
    if rand(1)<exp(-diff/temperatur)
        hlama=hbaru;
        zlama=zbaru;
        zrlama=zrbaru;
    end
    siklus=siklus+1;
    iterasi=iterasi+1;
end

end

grafik=[grafik zlama];

end
plot(grafik, '-');
h=hlama;

```

```

z=zlama;
zr=zrlama;
pl_pagi=zr(1);
pl_sore=zr(2);
pl_malam=zr(3);
pl_libur=zr(4);
pl_sorepagi=zr(5);
pl_pagimalam=zr(6);
pl_offonoff=zr(7);
waktukomputasi=cputime-waktukomputasi;
%%%cek solusi baru
function h=ceksolusi(h,ke)
if mod(ke,2)==1
    a=find(h(:,ke)==3);
    b=find(h(:,ke+1)==3);
    for s=1:length(a)
        for t=1:length(b)
            if a(s)==b(t)
                a(s)=0;
                b(t)=0;
            end
        end
    end
    hapusa=find(a(:,1)==0);
    hapusb=find(b(:,1)==0);
    a(hapusa,:)=[];
    b(hapusb,:)=[];
    for i=1:length(a)
        if a(i)~=b(i)
            p=a(i);
            q=b(i);
            %if p<=t || q<=t
            temph=h;
            h([p q],ke) = temph([q p],ke);
            %end
        end
    end
    if ke>=3 && ke~=30
        d=find(h(:,ke-1)==3);
        c=find(h(:,ke)==4);
        for s=1:length(c)
            for t=1:length(d)
                if c(s)==d(t)
                    c(s)=0;
                    d(t)=0;
                end
            end
        end
        hapusc=find(c(:,1)==0);
        hapusd=find(d(:,1)==0);
        c(hapusc,:)=[];
        d(hapusd,:)=[];
        for j=1:length(d)

```

```

        if c(j)~=d(j)
            u=c(j);
            v=d(j);
            temph=h;
            h([u v],ke) = temph([v u],ke);
        end
    end
end

if mod(ke,2)==0
    e=find(h(:,ke)==3);
    f=find(h(:,ke-1)==3);
    for s=1:length(e)
        for t=1:length(f)
            if e(s)==f(t)
                e(s)=0;
                f(t)=0;
            end
        end
    end
    hapuse=find(e(:,1)==0);
    hapusf=find(f(:,1)==0);
    e(hapuse,:)=[];
    f(hapusf,:)=[];
    for i=1:length(e)
        if e(i)~=f(i)
            p=e(i);
            q=f(i);
            temph=h;
            h([p q],ke) = temph([q p],ke);
        end
    end
end

%%%Pencarian fungsi objektif
function [z,zr]=obj(h,wp,ws,wm,wl,hari,perawat)
pl_pagi=0;
pl_sore=0;
pl_malam=0;
pl_libur=0;
for i=1:perawat
    %workload pagi sore malam libur
    cek_pagi=find(h(i,:)==1);
    wl_pagi=length(cek_pagi);
    cek_sore=find(h(i,:)==2);
    wl_sore=length(cek_sore);
    cek_malam=find(h(i,:)==3);
    wl_malam=length(cek_malam);
    cek_libur=find(h(i,:)==4);
    wl_libur=length(cek_libur);

    pl_pagi=pl_pagi + (abs(wl_pagi-wp)*3);

```

```

        pl_sore=pl_sore + (abs(wl_sore-ws)*3);
        pl_malam=pl_malam + (abs(wl_malam-wm)*3);
        pl_libur=pl_libur + (abs(wl_libur-wl)*3);
        pl=[pl_pagi pl_sore pl_malam pl_libur];
    end

    h(:,hari+1:hari+2)=0;
    %pelanggaran setelah shift sore diikuti shift pagi
    jum_pl_sorepagi=0;
    pl_sorepagi=0;
    for i=1:perawat
        for j=1:hari
            if h(i,j)==2 && h(i,j+1)==1
                jum_pl_sorepagi=jum_pl_sorepagi + 1;
                pl_sorepagi=jum_pl_sorepagi*4;
            end
        end
    end

    %pelanggaran setelah shift pagi diikuti shift malam
    jum_pl_pagimalam=0;
    pl_pagimalam=0;
    for i=1:perawat
        for j=1:hari
            if h(i,j)==1 && h(i,j+1)==3
                jum_pl_pagimalam=jum_pl_pagimalam + 1;
                pl_pagimalam=jum_pl_pagimalam*2;
            end
        end
    end

    jum_pl_offonoff=0;
    pl_offonoff=0;
    for i=1:perawat
        for j=1:hari
            if h(i,j)==4 && (h(i,j+1)==1 || h(i,j+1)==2) && h(i,j+2)==4
                jum_pl_offonoff=jum_pl_offonoff + 1;
                pl_offonoff=jum_pl_offonoff*2;
            end
        end
    end

    z=(sum(pl))+pl_sorepagi+pl_pagimalam+pl_offonoff;
    zr=[pl pl_sorepagi pl_pagimalam pl_offonoff];

```

LAMPIRAN B

Hasil Eksperimen

Pada lampiran ini akan diberikan hasil eksperimen yang telah dilakukan pada penelitian, yaitu hasil dari metode eksak, metode *simulated annealing*, dan kondisi eksisting.

Hasil eksperimen dengan metode eksak

Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	2	2	2	3	3	4	1	2	4	1	1	2	1	1	4	1	1	1	2	3	3	4	1	2	2	3	3	4	4
2	2	2	3	3	4	1	1	4	3	3	4	1	1	1	2	2	2	4	1	2	2	4	1	1	2	4	3	3	4	1
3	2	2	2	2	3	3	4	1	1	1	2	4	2	2	2	4	3	3	4	4	3	3	4	1	1	1	1	4	1	1
4	4	1	2	2	3	3	4	1	3	3	4	4	3	3	4	1	2	2	2	2	2	4	1	1	1	1	2	4	1	1
5	4	4	1	2	2	2	3	3	4	2	2	2	3	3	4	1	1	4	1	1	1	4	1	1	2	2	3	3	4	1
6	2	2	2	2	3	3	4	4	1	1	2	4	1	1	2	2	2	4	3	3	4	1	1	4	1	1	1	4	3	3
7	4	1	1	2	3	3	4	1	2	4	2	2	2	4	1	1	2	4	1	1	2	4	3	3	4	1	1	2	3	3
8	4	2	2	2	2	2	3	3	4	1	2	4	1	1	1	4	3	3	4	1	1	2	3	3	4	1	2	4	1	1
9	4	4	3	3	4	2	3	3	4	1	2	2	3	3	4	1	2	1	1	1	2	4	1	1	2	4	1	1	2	2
10	2	4	1	1	2	4	3	3	4	1	2	4	1	1	2	2	2	2	3	3	4	1	1	1	2	4	3	3	4	1

11	2	2	3	3	4	2	2	2	2	4	1	4	3	3	4	1	2	4	3	3	4	4	1	1	2	1	1	1	1	1
-----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	3	3	4	4	3	3	4	4	2	2	3	3	4	4	1	1	2	2	2	4	1	1	2	2	2	1	1	1	1	1
13	4	1	1	1	2	2	2	4	2	2	2	4	3	3	4	1	1	1	2	4	3	3	4	1	1	4	1	2	3	3
14	4	2	2	4	3	3	4	1	1	1	2	2	2	4	1	1	1	2	3	3	4	1	1	4	1	2	2	4	3	3
15	1	1	1	4	2	2	2	2	2	4	1	2	3	3	4	4	1	2	3	3	4	4	1	2	3	3	4	1	1	1
16	1	4	1	1	1	1	2	4	1	2	3	3	4	4	1	1	1	2	2	2	2	4	3	3	4	2	2	4	3	3
17	4	1	1	4	3	3	4	1	2	2	2	2	2	4	1	1	2	2	3	3	4	4	3	3	4	1	1	2	1	1
18	2	4	1	2	3	3	4	4	3	3	4	4	1	1	1	1	1	2	2	2	2	2	3	3	4	4	1	1	1	1
19	4	4	2	2	2	2	2	4	3	3	4	1	3	3	4	1	1	2	3	3	4	2	2	4	1	1	1	1	1	1
20	4	1	1	1	1	2	3	3	4	1	1	2	2	4	2	2	2	4	2	2	3	3	4	4	3	3	4	1	1	1
21	2	4	1	1	1	1	1	2	3	3	4	2	2	2	3	3	4	4	2	2	3	3	4	4	1	1	2	4	1	1
22	4	2	2	4	1	1	1	1	2	2	2	4	1	1	1	2	3	3	4	1	3	3	4	4	2	2	3	3	4	1
23	4	4	2	4	1	1	1	2	3	3	4	2	3	3	4	1	1	1	1	2	3	3	4	1	1	2	2	2	2	4
24	2	2	3	3	4	1	2	4	2	2	3	3	4	4	2	2	3	3	4	4	1	1	2	4	1	1	1	1	1	1
25	2	4	3	3	4	1	1	4	2	2	3	3	4	4	2	2	2	4	2	2	3	3	4	1	1	1	1	1	1	1
26	2	4	1	2	2	2	3	3	4	1	3	3	4	1	2	2	2	4	3	3	4	1	1	1	1	1	2	4	1	1
27	2	4	1	1	1	1	2	4	1	2	3	3	4	4	3	3	4	2	2	4	2	4	1	1	1	1	2	2	3	3
28	2	4	1	2	3	3	4	4	1	1	1	4	2	2	2	2	3	3	4	2	3	3	4	4	1	1	1	1	1	2
29	4	1	1	1	1	2	3	3	4	4	1	1	2	4	2	2	3	3	4	2	2	2	2	4	3	3	4	1	1	1
30	4	1	1	1	2	4	3	3	4	1	1	2	1	1	2	4	3	3	4	2	2	4	3	3	4	1	1	2	2	2

31	1	1	2	4	3	3	4	4	1	2	3	3	4	2	3	3	4	1	1	1	2	4	2	2	2	4	1	1	1	2
32	4	1	1	4	2	1	1	1	2	4	1	2	2	2	3	3	4	1	1	2	3	3	4	2	3	3	4	1	2	4
33	1	1	2	4	3	3	4	4	1	1	1	1	1	1	1	4	2	2	2	2	2	4	3	3	4	4	2	2	3	3
34	1	1	1	4	1	1	2	2	2	4	3	3	4	2	2	2	3	3	4	1	1	1	2	4	3	3	4	4	1	2
Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
35	4	1	1	4	1	1	2	2	2	2	3	3	4	1	1	4	1	1	2	4	3	3	4	2	3	3	4	1	2	2
36	3	3	4	1	2	4	1	1	1	2	2	4	1	1	2	2	3	3	4	1	1	4	3	3	4	1	2	2	2	4
37	4	1	2	2	2	4	3	3	4	1	1	2	2	2	2	4	1	1	1	1	1	4	1	2	3	3	4	4	3	3
38	1	4	3	3	4	1	2	4	1	2	2	4	1	1	2	4	2	4	1	1	1	1	2	2	3	3	4	2	3	3
39	1	1	1	4	3	3	4	2	3	3	4	1	2	2	2	4	1	1	2	2	2	2	1	4	3	3	4	1	1	4
40	3	3	4	1	2	4	3	3	4	1	1	1	1	4	1	1	1	4	1	2	2	2	2	4	2	2	3	3	4	2
41	3	3	4	1	2	4	3	3	4	4	1	1	2	2	3	3	4	1	1	1	1	4	1	1	2	4	2	2	2	2
42	3	3	4	1	2	4	3	3	4	1	1	1	1	1	2	4	1	2	2	4	1	1	2	2	2	4	3	3	4	2
43	1	4	1	1	2	4	2	2	3	3	4	1	1	4	2	2	3	3	4	1	1	4	1	2	2	4	1	2	3	3
44	4	1	2	4	2	2	3	3	4	4	1	1	1	2	3	3	4	1	2	2	2	4	1	1	1	2	3	3	4	1
45	4	1	1	4	2	2	3	3	4	1	1	4	1	1	1	1	2	2	2	2	2	4	3	3	4	2	3	3	4	1
46	3	3	4	1	2	4	1	2	2	4	3	3	4	1	1	1	1	1	2	2	2	4	3	3	4	4	1	1	2	2
47	1	2	3	3	4	4	3	3	4	1	2	4	1	2	2	2	2	4	1	1	1	1	1	4	3	3	4	1	2	2
48	1	1	3	3	4	4	1	1	1	4	2	4	2	4	1	2	2	2	2	4	3	3	4	1	1	1	2	2	3	3
49	1	2	2	4	3	3	4	1	3	3	4	1	1	4	1	1	1	1	2	2	2	4	1	4	3	3	4	2	2	2
50	2	4	3	3	4	4	1	2	2	4	3	3	4	4	1	1	1	1	1	1	1	1	2	2	3	3	4	2	2	2
51	3	3	4	1	1	4	2	2	2	4	1	1	3	3	4	2	3	3	4	1	1	1	1	4	1	2	2	4	2	2
52	3	3	4	1	2	4	1	1	1	4	3	3	4	1	3	3	4	2	2	4	2	2	2	1	1	1	1	4	2	2

53	1	2	3	3	4	4	1	1	1	4	3	3	4	2	3	3	4	1	2	4	1	1	2	4	1	1	2	2	2	2
54	3	3	4	1	1	1	1	2	3	3	4	1	2	4	3	3	4	1	2	4	2	2	2	2	2	4	1	1	1	4
55	2	4	1	2	3	3	4	4	1	1	1	1	1	4	1	1	2	4	3	3	4	2	2	1	2	4	2	2	3	3
56	1	4	2	2	2	4	1	1	1	1	1	1	2	4	3	3	4	4	2	2	3	3	4	1	2	2	3	3	4	1
57	1	4	3	3	4	4	2	2	2	4	1	1	2	4	3	3	4	1	1	1	1	1	1	1	2	3	3	4	2	2
Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
58	1	1	2	4	1	1	1	1	1	4	1	1	2	4	2	2	2	2	2	4	3	3	4	4	3	3	4	2	3	3
59	2	2	3	3	4	1	2	2	2	4	1	1	1	4	3	3	4	1	1	1	2	4	1	2	3	3	4	4	1	2
60	3	3	4	4	3	3	4	1	1	1	2	4	2	2	2	4	1	2	3	3	4	2	2	1	1	4	1	1	1	2
61	3	3	4	4	2	2	2	4	1	1	1	1	1	2	3	3	4	4	1	1	2	2	3	3	4	4	1	1	2	2
62	1	4	3	3	4	1	2	4	2	2	2	2	3	3	4	2	2	4	1	1	1	1	1	1	1	4	3	3	4	2
63	1	1	2	4	3	3	4	4	1	1	1	1	1	4	2	2	2	4	1	1	2	2	3	3	4	2	3	3	4	2
64	1	1	1	4	1	1	2	4	2	2	2	4	3	3	4	1	2	4	3	3	4	1	1	1	2	2	3	3	4	2
65	2	2	3	3	4	1	1	4	1	1	1	1	2	2	3	3	4	4	3	3	4	1	1	4	1	2	2	4	2	2
66	1	1	2	4	1	2	2	4	1	1	3	3	4	1	2	4	1	1	1	4	3	3	4	2	3	3	4	2	2	2
67	1	1	1	1	1	4	2	1	2	4	3	3	4	1	1	4	3	3	4	1	2	2	2	4	2	2	3	3	4	2
68	3	3	4	1	2	4	1	1	2	2	2	4	1	2	2	4	1	1	1	4	3	3	4	1	1	2	3	3	4	2
69	3	3	4	2	2	4	1	1	2	4	3	3	4	1	2	4	1	1	2	4	1	1	1	4	1	2	2	2	3	3
70	3	3	4	1	1	4	1	1	1	2	3	3	4	2	3	3	4	2	2	4	2	2	2	4	1	1	2	4	1	1
71	1	2	3	3	4	2	1	2	2	4	3	3	4	2	2	4	1	1	1	4	1	1	1	4	2	2	3	3	4	1
72	1	2	3	3	4	2	2	2	3	3	4	1	2	4	1	1	1	4	3	3	4	1	2	4	1	1	1	4	2	2
73	3	3	4	4	3	3	4	1	2	4	2	2	2	4	1	1	1	2	2	4	1	1	1	1	2	2	3	3	4	1
74	1	4	2	2	3	3	4	2	2	2	3	3	4	1	2	4	1	1	1	4	1	1	1	4	3	3	4	1	2	2

75	2	4	1	1	2	4	1	1	1	1	1	2	3	3	4	1	2	4	3	3	4	1	2	2	3	3	4	2	2	4
76	3	3	4	4	1	2	2	2	3	3	4	4	1	1	1	2	2	4	1	1	1	1	2	2	3	3	4	4	1	2
77	4	4	1	1	1	4	1	1	1	2	2	2	3	3	4	1	2	2	2	4	1	2	3	3	4	1	2	4	3	3
78	3	3	4	4	1	1	1	1	1	2	3	3	4	2	2	4	1	1	1	4	1	2	3	3	4	4	2	2	2	2
79	1	4	1	2	3	3	4	4	2	2	3	3	4	1	1	2	2	4	1	1	2	4	1	1	1	2	3	3	4	2
80	3	3	4	1	1	1	2	4	1	1	1	1	1	2	2	4	3	3	4	4	2	2	3	3	4	2	2	4	1	2
Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
81	1	1	2	4	1	1	1	1	2	2	3	3	4	1	2	4	3	3	4	1	2	4	3	3	4	1	2	4	2	2
82	1	2	2	4	1	1	1	2	3	3	4	4	3	3	4	4	3	3	4	1	2	2	2	4	1	1	1	1	2	2
83	3	3	4	1	1	1	1	2	3	3	4	4	3	3	4	4	2	1	2	4	1	1	2	2	2	4	1	1	2	2
84	1	1	2	4	1	1	1	2	3	3	4	2	2	4	1	1	2	4	3	3	4	4	1	2	3	3	4	1	2	2
85	3	3	4	1	1	1	2	2	3	3	4	1	1	1	1	4	3	3	4	4	2	2	2	2	2	4	1	1	2	4

Hasil Eksperimen dengan *simulated annealing* dengan solusi yang terbaik

Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	4	2	2	3	3	4	2	2	1	1	1	1	1	1	1	2	2	2	1	1	1	2	2	3	3	4	1	2	2
2	2	4	1	1	1	2	2	2	2	4	2	4	1	1	1	1	3	3	4	4	1	4	1	2	2	2	2	4	3	3
3	4	1	1	1	2	1	1	2	2	2	1	4	3	3	4	2	2	4	1	2	1	4	1	2	3	3	4	4	3	3
4	3	3	4	1	2	2	3	3	4	4	1	2	3	3	4	1	2	4	1	1	2	2	3	3	4	1	1	1	2	2
5	3	3	4	2	3	3	4	1	2	4	1	1	1	2	3	3	4	2	2	2	2	1	1	1	1	2	3	3	4	4
6	2	2	2	4	2	1	1	2	3	3	4	2	2	2	2	4	1	4	1	4	2	4	1	1	3	3	4	1	1	1
7	1	2	3	3	4	1	2	1	1	2	3	3	4	1	1	1	2	4	2	1	1	4	2	4	3	3	4	4	2	2
8	2	4	1	2	2	4	3	3	4	1	1	1	1	2	1	1	3	3	4	2	3	3	4	4	2	2	2	4	1	1
9	2	2	3	3	4	4	3	3	4	2	1	1	2	2	2	1	1	1	1	4	2	4	2	1	3	3	4	4	1	1
10	3	3	4	2	3	3	4	4	1	1	1	1	1	4	1	1	2	2	2	2	2	4	3	3	4	4	2	2	3	3
11	1	1	1	1	2	4	2	2	2	2	1	4	3	3	4	2	3	3	4	1	1	2	2	4	3	3	4	1	1	4
12	1	4	2	2	2	4	2	2	2	2	2	4	1	1	2	4	1	4	3	3	4	1	1	4	3	3	4	1	1	1
13	2	2	2	4	1	1	1	2	3	3	4	4	1	1	2	4	1	4	3	3	4	2	2	4	1	1	2	4	1	1
14	4	4	3	3	4	1	1	2	2	2	1	1	1	2	3	3	4	1	1	4	3	3	4	1	2	2	3	3	4	2
15	1	2	3	3	4	4	2	4	1	4	2	2	2	4	1	4	1	1	1	2	2	2	2	4	1	1	1	1	1	4
16	2	2	1	1	1	2	3	3	4	2	3	3	4	1	1	4	3	3	4	2	2	2	3	3	4	4	1	1	2	4
17	4	1	2	4	3	3	4	1	1	2	3	3	4	2	3	3	4	1	2	2	1	2	2	4	1	1	1	4	2	1
18	3	3	4	2	2	4	1	1	3	3	4	4	1	1	1	4	1	2	2	2	2	1	2	2	3	3	4	1	1	4
19	1	2	2	2	2	4	3	3	4	1	2	2	1	2	3	3	4	1	3	3	4	4	3	3	4	1	1	1	1	1
20	2	2	3	3	4	4	2	2	2	2	2	4	1	4	1	1	2	4	2	4	3	3	4	1	1	1	1	1	1	1

21	1	1	2	4	1	1	2	2	2	4	3	3	4	1	1	2	3	3	4	4	3	3	4	2	3	3	4	1	2	2	
Perawat	Hari																														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
22	1	1	3	3	4	1	2	1	1	4	2	4	2	4	1	1	2	2	3	3	4	1	1	2	3	3	4	2	3	3	
23	4	1	1	1	2	4	3	3	4	2	4	1	2	4	3	3	4	4	1	1	1	1	1	1	2	2	2	2	2	2	
24	1	1	1	4	1	2	2	4	1	2	2	4	1	4	2	2	3	3	4	1	2	2	3	3	4	1	1	4	3	3	
25	1	1	1	2	2	2	3	3	4	4	1	1	3	3	4	1	2	4	2	4	1	1	1	1	2	2	2	4	2	4	
26	3	3	4	1	2	4	1	2	2	4	3	3	4	4	2	2	1	1	1	2	1	2	1	1	2	1	3	3	4	4	
27	2	4	1	1	2	2	2	2	3	3	4	4	1	4	1	1	1	1	3	3	4	1	2	4	1	1	1	2	2	4	
28	1	2	2	4	1	2	3	3	4	1	1	2	2	4	1	4	2	4	3	3	4	1	1	2	3	3	4	2	1	1	
29	1	4	2	2	2	2	3	3	4	1	1	2	1	2	3	3	4	1	1	1	2	4	3	3	4	4	3	3	4	1	
30	1	1	1	1	1	1	1	2	2	4	1	2	2	2	2	4	3	3	4	2	2	4	3	3	4	4	3	3	4	1	
31	2	4	3	3	4	2	3	3	4	4	2	2	3	3	4	1	1	2	2	1	1	1	1	1	1	2	2	4	1	4	
32	1	1	2	4	2	2	2	2	3	3	4	1	1	4	4	2	3	3	4	1	3	3	4	1	1	2	1	1	2	4	
33	3	3	4	2	3	3	4	2	3	3	4	4	2	4	1	1	1	2	1	1	1	1	2	2	2	4	3	3	4	1	
34	3	3	4	1	1	1	1	1	1	1	1	2	3	3	4	4	3	3	4	4	1	2	2	2	2	2	3	3	4	2	
35	3	3	4	4	3	3	4	1	3	3	4	2	2	4	1	2	1	1	1	2	2	2	2	1	1	2	1	1	4	2	4
36	1	4	1	2	2	4	1	1	2	4	1	1	1	1	2	4	2	4	3	3	4	4	2	2	2	2	3	3	4	1	
37	3	3	4	1	2	2	3	3	4	1	1	2	3	3	4	1	1	2	2	4	1	1	1	4	1	2	2	4	2	4	
38	1	4	3	3	4	2	4	1	1	1	1	2	2	4	1	1	1	4	2	4	2	4	1	2	3	3	4	2	2	2	
39	1	1	3	3	4	1	1	4	1	1	2	4	1	1	2	2	2	2	2	4	1	4	1	4	2	4	1	2	3	3	
40	1	2	3	3	4	2	3	3	4	1	2	2	2	2	3	3	4	1	2	2	3	3	4	4	1	1	1	1	1	1	
41	3	3	4	2	2	1	1	1	2	2	1	1	3	3	4	2	2	4	3	3	4	1	1	2	3	3	4	4	1	4	
42	1	2	3	3	4	4	3	3	4	2	2	2	2	4	3	3	4	1	1	2	3	3	4	2	2	4	1	1	1	1	

43	4	1	1	1	1	2	3	3	4	1	2	2	3	3	4	1	1	4	1	1	4	2	3	3	4	2	2	2	3	3
44	3	3	4	1	1	2	1	4	2	2	2	1	1	2	2	4	1	2	3	3	4	4	1	1	2	4	1	4	3	3
Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
45	3	3	4	1	1	1	2	4	1	1	2	2	2	4	1	1	3	3	4	2	4	1	2	2	3	3	4	4	3	3
46	4	2	2	4	3	3	4	1	1	1	1	1	2	2	3	3	4	2	3	3	4	1	1	2	3	3	4	1	1	2
47	4	1	2	1	1	1	2	2	2	2	4	1	1	1	2	4	2	4	1	1	4	4	1	1	1	4	1	2	3	3
48	3	3	4	1	1	2	3	3	4	1	1	1	2	2	3	3	4	1	1	1	4	4	2	2	3	3	4	1	2	4
49	2	4	1	1	1	1	1	1	1	2	3	3	4	4	2	2	3	3	4	4	3	3	4	1	2	2	3	3	4	1
50	1	1	1	1	1	1	2	4	2	2	3	3	4	2	3	3	4	1	1	4	3	3	4	4	1	2	3	3	4	2
51	2	2	3	3	4	4	1	2	3	3	4	1	2	2	2	2	1	2	3	3	4	1	3	3	4	1	1	1	1	1
52	4	1	1	1	3	3	4	4	1	1	3	3	4	2	2	2	3	3	4	1	4	2	2	2	2	4	1	1	1	2
53	2	2	2	2	2	1	1	4	3	3	4	2	3	3	4	1	1	4	3	3	4	1	3	3	4	1	1	2	2	4
54	3	3	4	4	3	3	4	1	1	1	1	4	3	3	4	1	2	2	2	2	2	4	3	3	4	1	2	2	2	1
55	2	2	2	2	3	3	4	4	1	1	3	3	4	1	1	4	2	2	2	4	1	4	2	4	1	1	1	1	3	3
56	1	2	3	3	4	1	1	1	3	3	4	1	1	1	2	4	3	3	4	1	2	4	3	3	4	2	2	2	2	4
57	4	1	2	2	2	2	2	4	1	1	2	2	3	3	4	4	1	4	1	1	1	4	1	1	1	4	1	2	3	3
58	2	4	3	3	4	2	3	3	4	4	3	3	4	1	1	1	1	1	2	2	3	3	4	1	1	1	1	2	2	2
59	4	1	2	2	3	3	4	1	3	3	4	2	2	4	3	3	4	1	1	1	2	1	2	2	2	4	1	1	1	4
60	1	1	1	2	2	4	2	4	2	2	2	4	1	1	1	2	4	1	2	4	1	2	3	3	4	1	3	3	4	1
61	3	3	4	2	3	3	4	1	2	4	1	1	2	2	2	2	1	1	2	4	1	1	3	3	4	2	3	3	4	1
62	4	4	2	4	1	1	1	1	2	2	3	3	4	1	1	4	1	1	2	2	3	3	4	4	2	4	1	1	2	2
63	1	1	1	1	1	2	2	4	2	4	2	2	2	4	2	4	4	2	3	3	4	1	1	1	3	3	4	2	2	1
64	2	2	2	4	1	4	1	4	2	4	3	3	4	1	3	3	4	1	2	4	4	1	1	2	2	4	1	1	1	2

65	2	2	2	2	2	4	2	4	1	1	2	4	1	1	3	3	4	1	1	1	4	4	2	4	1	2	3	3	4	1	
66	2	4	1	1	1	4	3	3	4	1	1	4	2	4	1	1	1	2	4	1	2	4	2	2	2	2	3	3	4	2	
67	4	1	3	3	4	4	2	1	1	2	3	3	4	1	2	2	4	1	4	4	1	2	2	1	1	1	2	2	3	3	
Perawat	Hari																														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
68	4	1	1	2	2	1	1	1	3	3	4	1	2	2	3	3	4	2	4	2	3	3	4	1	3	3	4	1	1	2	
69	3	3	4	4	1	1	1	1	1	2	3	3	4	4	3	3	4	1	1	1	4	2	3	3	4	1	2	2	2	2	
70	1	2	2	4	1	2	2	2	3	3	4	1	1	2	4	2	2	4	4	1	1	1	4	1	1	1	1	4	3	3	
71	4	1	1	1	1	1	3	3	4	1	2	4	2	4	2	2	3	3	4	4	1	1	3	3	4	2	2	2	3	3	
72	2	2	2	1	2	4	2	4	2	4	1	1	1	1	1	1	4	2	3	3	4	4	1	1	1	2	3	3	4	1	
73	2	4	1	1	1	2	2	4	1	1	2	2	2	4	2	2	4	1	4	4	4	4	4	1	1	1	4	2	2	3	3
74	2	4	3	3	4	4	1	1	2	2	2	2	3	3	4	4	3	3	4	1	2	2	1	1	1	1	2	1	1	4	
75	2	4	1	4	1	2	3	3	4	1	2	2	3	3	4	1	1	1	1	4	3	3	4	2	3	3	4	1	1	2	
76	1	2	2	4	1	2	1	2	3	3	4	1	1	1	3	3	4	1	1	2	2	4	3	3	4	1	2	4	2	4	
77	1	2	3	3	4	1	2	2	2	2	3	3	4	4	3	3	4	2	1	1	1	4	3	3	4	2	2	4	1	1	
78	2	4	1	1	1	1	1	4	1	4	2	2	4	1	1	2	2	4	3	3	4	2	2	4	1	1	2	2	3	3	
79	2	2	4	2	4	1	1	1	1	2	3	3	4	1	4	1	1	4	1	4	1	1	4	2	2	4	2	2	2	2	
80	3	3	4	2	3	3	4	1	1	1	2	4	2	1	2	2	2	4	1	1	1	2	4	4	1	2	3	3	4	1	
81	1	2	3	3	4	2	2	2	2	4	4	1	2	4	1	1	2	4	1	2	3	3	4	4	1	1	2	4	1	1	
82	4	1	1	2	3	3	4	1	1	2	2	2	4	2	3	3	4	2	3	3	4	1	1	1	3	3	4	2	4	1	
83	3	3	4	2	3	3	4	1	3	3	4	1	4	1	2	2	2	4	1	1	2	4	2	4	1	1	2	2	1	1	
84	4	1	1	1	2	2	2	4	1	1	1	1	1	2	4	4	4	2	2	2	3	3	4	2	4	4	3	3	4	1	
85	4	1	3	3	4	1	1	2	3	3	4	4	3	3	4	1	1	4	2	4	1	2	2	1	1	2	2	2	2	4	

Penjadwalan pada kondisi eksisting

Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	1	1	1	1	2	2	4	3	3	4	4	1	1	1	2	4	1	3	3	4	4	1	1	1	2	4	1	2	2
2	3	3	4	4	1	1	2	2	2	4	3	3	4	4	1	1	1	1	3	3	4	4	1	1	2	2	2	4	3	3
3	1	2	2	2	4	1	3	3	4	1	1	1	2	2	3	3	4	4	1	2	2	2	4	1	3	3	4	1	1	1
4	1	2	2	2	4	1	3	3	4	1	1	1	2	2	3	3	4	4	1	2	2	2	4	1	3	3	4	1	1	1
5	2	4	3	3	4	1	1	1	2	2	2	4	3	3	4	4	2	2	2	4	3	3	4	1	1	1	2	2	2	4
6	3	3	4	4	1	1	1	2	2	2	4	1	3	3	4	1	2	2	3	3	4	4	1	1	1	2	2	2	4	1
7	2	4	3	3	4	1	1	1	2	2	2	4	3	3	4	4	2	2	2	4	3	3	4	1	1	1	2	2	2	4
8	1	1	4	1	1	1	1	1	1	1	4	4	1	1	1	1	4	4	1	1	4	1	1	1	1	1	1	1	4	4
9	4	4	2	2	2	2	2	4	3	3	4	4	2	2	2	2	3	3	4	2	2	2	2	4	3	3	4	4	2	2
10	3	3	4	4	1	1	1	2	3	3	4	2	1	1	2	2	3	3	4	4	1	1	1	2	3	3	4	2	1	1
11	3	3	4	4	1	1	1	2	3	3	4	2	1	1	2	2	3	3	4	4	1	1	1	2	3	3	4	2	1	1
12	2	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	1	4	1	3	3	4	4	1	1	2	2	3	3
13	2	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	1	4	1	3	3	4	4	1	1	2	2	3	3
14	2	2	2	4	3	3	4	4	1	1	1	2	2	4	3	3	4	4	2	2	4	1	3	3	4	4	1	1	3	3
15	3	3	4	4	3	3	4	4	2	2	2	2	2	2	2	4	3	3	4	4	3	3	4	4	2	2	2	2	2	2
16	4	1	1	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	4	3	3	4	4	1	1	1	2	2	2
17	4	4	1	2	2	2	4	1	3	3	4	1	1	2	2	2	3	3	4	4	4	1	2	2	2	4	1	3	3	4
18	3	3	4	4	1	2	2	4	3	3	4	4	1	1	1	2	2	2	4	1	2	2	3	3	4	4	2	1	1	1
19	1	2	2	2	3	3	4	4	4	1	1	1	2	2	2	4	3	3	4	1	1	1	2	2	2	4	3	3	4	1
20	4	1	1	2	2	2	2	4	3	3	4	4	3	3	4	1	2	2	2	4	1	1	2	2	2	2	4	3	3	4

21	2	4	3	3	4	4	1	1	4	2	2	2	3	3	4	4	4	2	2	2	4	3	3	4	4	1	1	4	2	2	
Perawat	Hari																														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
22	2	4	3	3	4	4	1	1	4	2	2	2	3	3	4	4	2	2	3	3	4	4	1	1	4	2	2	2	3	3	
23	1	2	2	2	3	3	4	4	4	1	3	3	4	2	2	2	3	3	4	1	2	2	2	3	3	4	4	1	1	1	
24	3	3	4	4	1	1	2	2	2	4	3	3	4	4	2	2	1	1	1	1	3	3	4	4	1	1	2	2	2	4	
25	4	1	1	1	2	2	3	3	4	1	3	3	4	4	1	1	3	3	4	2	3	3	4	4	2	4	1	1	1	2	
26	4	1	1	1	2	2	2	4	3	3	4	4	1	1	3	3	4	4	1	1	1	2	2	2	4	3	3	4	4	1	
27	1	1	1	1	1	4	1	1	4	1	1	1	4	1	1	1	1	4	1	4	1	1	1	1	1	1	4	1	1	1	
28	4	4	4	1	4	1	2	2	3	3	4	4	4	1	2	2	3	3	4	4	1	1	1	4	2	2	1	2	2	1	
29	4	1	1	1	2	2	4	4	4	4	1	4	4	1	4	4	4	4	4	4	4	4	2	2	3	3	4	4	1	1	
30	1	1	2	2	3	3	3	3	4	1	1	4	2	2	3	3	4	4	4	1	2	2	2	1	1	4	4	1	2	2	
31	3	3	4	4	4	4	1	1	2	2	3	3	4	4	1	1	2	2	3	3	4	4	3	3	4	4	3	3	4	4	
32	2	2	3	3	4	4	4	4	4	1	2	2	3	3	4	4	1	4	2	2	3	3	4	4	1	4	2	2	3	3	
33	1	1	1	1	1	4	1	1	1	1	1	4	1	1	1	1	1	1	4	4	1	1	1	1	1	1	1	4	1	1	4
34	3	3	4	4	4	4	2	2	3	3	4	4	1	1	1	1	1	1	2	2	3	3	4	4	1	4	4	4	4	2	2
35	4	4	4	4	2	2	3	3	4	4	1	4	4	1	4	4	2	2	3	3	4	4	1	1	4	4	4	1	4	4	
36	2	2	3	3	4	4	4	1	2	2	3	3	4	4	2	1	1	4	4	4	2	2	3	3	4	4	1	1	1	1	
37	4	4	2	2	3	3	4	4	4	1	1	1	4	4	3	2	3	3	4	4	1	1	1	1	1	4	2	2	3	3	
38	4	2	2	2	1	4	1	1	1	1	1	4	2	2	3	3	4	4	4	1	1	1	1	1	2	2	3	3	4	4	
39	1	1	1	1	4	1	1	1	4	1	2	2	3	3	4	4	1	4	1	4	1	1	1	2	3	3	4	4	4	4	
40	3	3	4	4	4	1	1	2	3	3	4	4	1	1	1	2	3	3	4	4	4	1	2	2	3	3	4	4	4	1	
41	4	1	1	4	1	1	1	1	1	1	4	1	1	1	4	4	1	1	4	1	1	1	1	1	1	4	1	1	1	4	
42	2	2	3	3	4	4	1	1	1	4	2	2	3	3	4	2	3	3	4	4	2	1	1	1	4	2	2	3	3	4	

43	3	3	4	4	1	1	2	2	2	3	4	4	1	1	2	2	1	1	4	1	1	2	2	2	3	4	4	1	1	2	
44	4	1	1	4	2	2	3	3	4	4	1	1	2	2	3	3	4	4	4	2	2	1	3	3	4	1	1	2	2	3	
Perawat	Hari																														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
45	4	4	2	2	3	3	4	4	4	2	3	3	4	4	4	4	4	2	2	3	3	4	4	4	2	3	3	4	4	4	
46	1	1	4	1	1	1	1	1	2	4	4	1	1	1	1	1	1	4	1	1	1	1	1	2	4	4	1	1	1	1	
47	3	3	4	4	1	1	1	1	1	2	3	3	4	4	4	1	1	2	2	1	1	1	1	1	2	3	3	4	4	4	
48	4	1	4	4	1	1	1	1	2	4	4	1	1	1	4	4	1	4	4	1	1	1	1	2	4	4	1	1	1	4	
49	2	2	3	3	4	4	4	1	1	1	2	2	3	3	4	2	2	3	3	4	4	4	4	1	1	1	2	2	3	3	4
50	4	4	4	4	1	1	2	2	1	3	4	4	1	1	1	4	4	4	4	1	1	2	2	1	3	4	4	1	1	1	
51	4	1	4	1	2	2	3	3	4	4	1	1	1	1	2	4	1	4	1	2	2	3	3	4	4	1	1	1	1	2	
52	1	1	2	2	3	3	4	4	4	4	4	1	2	2	3	3	4	4	2	3	3	4	4	4	4	4	1	2	2	3	
53	1	2	3	3	4	4	1	4	4	2	3	3	4	4	4	1	3	3	4	4	2	1	4	4	2	3	3	4	4	4	
54	3	3	4	4	1	1	1	2	3	3	4	2	1	1	2	2	3	3	4	4	1	1	1	2	3	3	4	2	1	1	
55	3	3	4	4	1	1	1	2	3	3	4	2	1	1	2	2	3	3	4	4	1	1	1	2	3	3	4	2	1	1	
56	2	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	1	4	1	3	3	4	4	1	1	2	2	3	3	
57	2	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	1	4	1	3	3	4	4	1	1	2	2	3	3	
58	2	2	2	4	3	3	4	4	1	1	1	2	2	4	3	3	4	4	2	2	4	1	3	3	4	4	1	1	3	3	
59	3	3	4	4	3	3	4	4	2	2	2	2	2	2	2	4	3	3	4	4	3	3	4	4	2	2	2	2	2	2	
60	4	1	1	4	3	3	4	4	1	1	1	2	2	2	3	3	4	4	1	4	3	3	4	4	1	1	1	2	2	2	
61	4	4	1	2	2	2	4	1	3	3	4	1	1	2	2	2	3	3	4	4	4	1	2	2	2	4	1	3	3	4	
62	3	3	4	4	1	2	2	4	3	3	4	4	1	1	1	2	2	2	4	1	2	2	3	3	4	4	2	1	1	1	
63	1	2	2	2	3	3	4	4	4	1	1	1	2	2	2	4	3	3	4	1	1	1	2	2	2	4	3	3	4	1	
64	4	1	1	2	2	2	2	4	3	3	4	4	3	3	4	1	2	2	2	4	1	1	2	2	2	2	4	3	3	4	

65	2	2	3	3	4	4	4	1	1	1	2	2	3	3	4	2	2	3	3	4	4	4	1	1	1	2	2	3	3	4
66	4	4	4	4	1	1	2	2	1	3	4	4	1	1	1	4	4	4	4	1	1	2	2	1	3	4	4	1	1	1
67	4	1	4	1	2	2	3	3	4	4	1	1	1	1	2	4	1	4	1	2	2	3	3	4	4	1	1	1	1	2
Perawat	Hari																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
68	1	1	2	2	3	3	4	4	4	4	4	1	2	2	3	3	4	4	2	3	3	4	4	4	4	4	1	2	2	3
69	1	2	3	3	4	4	1	4	4	2	3	3	4	4	4	1	3	3	4	4	2	1	4	4	2	3	3	4	4	4
70	1	2	2	2	4	1	3	3	4	1	1	1	2	2	3	3	4	4	1	2	2	2	4	1	3	3	4	1	1	1
71	1	2	2	2	4	1	3	3	4	1	1	1	2	2	3	3	4	4	1	2	2	2	4	1	3	3	4	1	1	1
72	3	3	4	4	4	4	1	1	2	2	3	3	4	4	1	1	2	2	3	3	4	4	3	3	4	4	3	3	4	4
73	2	2	3	3	4	4	4	4	4	1	2	2	3	3	4	4	1	4	2	2	3	3	4	4	1	4	2	2	3	3
74	1	1	1	1	1	4	1	1	1	1	1	4	1	1	1	1	1	1	4	4	1	1	1	1	1	1	4	1	1	4
75	3	3	4	4	4	4	2	2	3	3	4	4	1	1	1	1	1	1	2	2	3	3	4	4	1	4	4	4	2	2
76	4	4	4	4	2	2	3	3	4	4	1	4	4	1	4	4	2	2	3	3	4	4	1	1	4	4	4	1	4	4
77	2	2	3	3	4	4	1	1	1	4	2	2	3	3	4	2	3	3	4	4	2	1	1	1	4	2	2	3	3	4
78	3	3	4	4	1	1	2	2	2	3	4	4	1	1	2	2	1	1	4	1	1	2	2	2	3	4	4	1	1	2
79	4	1	1	4	2	2	3	3	4	4	1	1	2	2	3	3	4	4	4	2	2	1	3	3	4	1	1	2	3	3
80	4	4	2	2	3	3	4	4	4	2	3	3	4	4	4	4	4	2	2	3	3	4	4	4	2	3	3	4	4	4
81	1	1	4	1	1	1	1	1	2	4	4	1	1	1	1	1	1	4	1	1	1	1	1	2	4	4	1	1	1	1
82	3	3	4	4	1	1	1	1	1	2	3	3	4	4	4	1	1	2	2	1	1	1	1	1	2	3	3	4	4	4
83	4	4	4	4	2	2	3	3	4	4	1	4	4	1	4	4	2	2	3	3	4	4	1	1	4	4	4	1	4	4
84	4	4	4	4	2	2	3	3	4	4	1	4	4	1	4	4	2	2	3	3	4	4	1	1	4	4	4	1	4	4
85	2	2	3	3	4	4	4	4	4	1	2	2	3	3	4	4	1	4	2	2	3	3	4	4	1	4	2	2	3	3

BIODATA PENULIS



Penulis bernama lengkap Aditya Pratama Hidayatullah dilahirkan di Surabaya 6 April 1992. Penuli merupakan anak pertama dari tiga saudara dengan orang tua bernama Lili Aliyudin dan Ernawati.

Penulis telah menempuh pendidikan formal mulai dari TK Tunas Bakti SBY, SDN Dr. Soetomo VII SBY, SMPN 3 SBY, SMAN 15 SBY, dan Teknik Industri Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Selama perkuliahan penulis aktif di berbagai kegiatan kemahasiswaan. Diantaranya menjadi staff Departemen KWU 11/12, Kepala Departemen KWU 12/13, Asisten Laboratorium Komputasi dan Optimasi Industri (KOI), dan Senat Mahasiswa 13/14.

Penulis juga pernah mengikuti kerja praktek di PT PJB Surabaya selama 1 bulan dimana pekerjaan disana adalah menganalisa komponen kritis pada Pembangkit Listrik Tenaga Uap (PLTU) di Unit Pembangkit (UP) Gresik. Pada periode Juli-Agustus 2013.

Penulis dapat dihubungi via email : adityaph92@gmail.com